Advances In Explainable Artificial Intelligence, Fair Machine Learning, And The Intersections Thereof

By

MICHAEL J. LIVANOS
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Ian Davidson, Chair

Hamed Pirsiavash

Jiawei Zhang

Committee in Charge

2024

Dedicated to my barking dog, Delilah

Contents

Abstract

Advances In Explainable Artificial Intelligence, Fair Machine Learning, And The Intersections Thereof

Artificial intelligence (AI), if used correctly, has the capacity to improve human life by automating procedures that previously required human expertise and precision, particularly those that may have a great impact on people's lives and where the cost of a mistake is high. Unfortunately, the use of machine learning (ML) algorithms carries with it certain risks that may limit their applicability in such sensitive domains. Particularly, ML algorithms solve tasks by optimizing a complex non-linear mapping between an input and output space. While the automated process of tuning this function is powerful, it ultimately renders these learners uninterpretable and subject to error, misuse, or harmful bias.

The fields of explainable artificial intelligence (XAI) and fair machine learning exist to combat these issues. XAI seeks to explain how ML agents operate in human-interpretable terms, while fairness aims to correct or avoid potential unfair outcomes. While existing work has laid promising groundwork toward these ends, there are several limitations in both domains that should be rectified before AI can be trusted for particularly sensitive tasks.

This dissertation aims to extend XAI and fair machine learning by making headway on these limitations. For XAI, we create approaches that explain the entire model, not just individual actions, we develop techniques tailored towards ML tasks beyond supervised learning, and we examine alternatives to input space as the means of providing that explanation. For fairness, we look to the literature in social sciences to create fair ML algorithms that match the models of how unfairness and discrimination occur, which we argue are superior to existing techniques that do not leverage this theory. Finally, we introduce the novel concept of machine-to-machine explanation: the idea that explanation technology can be used for additional computational tasks, enabling collaboration among ML models to improve their performance.

Acknowledgments

I would like to briefly thank all those who have supported me in my doctoral studies journey. Your incredible support over these last five years has inspired me tremendously. For better or for worse, I cannot assign Shapley values to all of your input, but to all those on this list, I hope you know that our time and friendship are irreplaceable.

First and foremost, I thank Professor Ian Davidson for his incredible mentorship over my Ph.D. career. Not only have you taught me how to conduct myself as an academic researcher, but your generosity has played a major role in helping me enjoy my time here. From the times that you've invited me to coffee or just offered to chat between meetings, I am reminded of how fortunate I am to have you as an advisor.

I would also like to thank those who served on my dissertation and qualifying exam committee. Beyond Professor Davidson, I thank Professor Hamed Pirsiavash, Professor Jiawei Zhang, Professor Joshua McCoy, and Professor Antoine Gourru. Your insightful commentary has been crucial in helping me hone the narrative of this dissertation.

I am also immensely grateful to my wife Josefina for her support and encouragement over the years. Thanks for ensuring that I never worked too hard and providing the emotional support I needed to finish this out. I hope that I can be as supportive of a partner to you as you are to me. There is no one else I would rather have gone through this journey with and I am excited to move on to the next stage of life with you.

I have also learned so much from my labmates and collaborators who provided me the feedback I needed to produce high-quality work. In particular, thank you to Kurt, Ge, Hongjing, and Zilong, all of whom in their positions as more senior researchers have taught me a lot about conducting high-quality research. Along with them, our insightful undergraduate friends Nicholas, Thomas, Sreya, Jessica, Kaoushik, Avigail, Scott, and Stephen - thank you for your commentary on my research as it progressed and the insight you bring to lab meetings.

I would be remiss if I did not mention those who helped me before coming to graduate school. Most notably, my former research advisors Professors Matthew Johnson and Edgar Peña - both of you gave me an incredible introduction to academic research and served as amazing role models for how to conduct myself as an advisor to undergraduates I have mentored.

Teaching has been one of the most transformative and enjoyable experiences I have had at UC Davis, and I would like to thank all of my students for putting such great effort into the classes I've taught. I wish you all the success and happiness in the world in your future endeavors.

Gracias a mi nueva familia, en particular a Jesús, Roxana, y mis suegros. Gracias por darme la bienvenida a su casa, su vida y sus corazones. Es un gran privilegio ser parte de esta familia.

I also extend my deepest gratitude to all of those who have helped me along my professional journey beyond academics, including my sister Melissa who patiently helped me prepare my essays for college and graduate school, my parents for their constant support of my academic endeavors, and last but not least Ashli who - well, she's done something. I'm sure of it.

I thank my former roommates and lifelong friends José, Aldo, Daniel, and Dakota for the years of fun we had together. My good friends Mario, Ben, and Nathan for being there for everything from personal milestones to bouncing off ideas to joking around. My SHPE familia who were the ones who introduced me to the idea of graduate school. I thank Gaby and Ricky, who have been such a kind and supportive force in my life. Sabrina, Rachel, Jasmine, and Lia, for their friendship and guidance over several years. Thank you for helping me smile on even the toughest of days. Lastly, I thank Ariana Grande for the literal thousands of hours of entertainment and comfort I have found in your music.

CHAPTER 1

Introduction

The field of artificial intelligence (AI) has proven to be incredibly powerful for many complex tasks from neural machine translation [1] to playing video games [2], often surpassing human abilities. Increasingly, researchers and practitioners alike have attempted to leverage the success of deep learning for highly sensitive, high-risk tasks that impact human lives, such as medical AI [3,4], autonomous vehicles [5], and criminal justice [6]. Despite strong performance in traditional AI tasks, the use of AI in these sensitive areas carries inherent risks. Humans typically approach tasks with multiple high-level goals, while machines optimize a single, complex non-linear function to meet specific mathematical objectives. This discrepancy leads to challenges, such as powerful neural networks failing to generalize to novel data [7, 8], latching onto spurious features in the training data [9], or having their objectives exploited by malicious actors [10].

Ensuring that the goals of machine learning algorithms align with those of humans is one of the major challenges of the field, often referred to as the alignment problem [11]. Unfortunately, many of the most powerful machine learning models are inherently black-box algorithms, meaning there is no obvious method of interrogating these algorithms to determine if such shortcomings will exist in any particular network. Consequently, research addressing the alignment problem has taken two major forms: explainable artificial intelligence (XAI) and fair machine learning.

The field of explainable artificial intelligence seeks to provide a layer of interpretability to these otherwise opaque models [12]. Interpretability is challenging to rigorously define, as what is interpretable to one person may not be interpretable to another. Existing work has approached this challenge in several ways [13] such as highlighting the features responsible for prediction [14, 15], creating inherently interpretable models [16], or providing greater context to how the model operates [17]. Explanations can be post-hoc, explaining an existing uninterpretable model [15, 17, 18], or in-situ, creating models that are inherently interpretable [10, 16, 19].

A similar but distinct field is fairness in machine learning, which seeks to mitigate potential biases that machine learning algorithms may exploit to achieve lower loss [20]. This is typically

done by interrogating an algorithm's output with respect to the protected status variable(s) (PSV) of the data [21] and a chosen fairness metric, of which many are used in the literature [22]. Fairness interventions address these concerns by post-processing the output of an unfair network to make it fairer, modifying aspects of the network to improve fairness in the future, or employing online methods to train models to be fair [23].

Many techniques for adding interpretability and increasing fairness existed in the literature prior to the work compiled in this dissertation. However, I argue that this work was either insufficient for handling the challenges they set out to address, or that there are unexplored aspects of these fields which, when exploited, have the potential to provide much more equitable systems. Acknowledging these limitations and extending these techniques, as done in this work, is essential for calibrating trust in machine learning models.

1.1. Limitations Of Existing Explainable AI

In 2019, a DARPA initiative sought to develop a series of critical questions that must be addressed before AI can be trusted in highly sensitive domains. Those questions are: "Why did [the model] do that?", "why not something else?", "When do[es the model] succeed/fail?" and "How can I correct a mistake" [24]. Existing work in XAI has primarily focused on the first two questions, often overlooking the others, and this dissertation aims to expand the exploration of these topics.

While existing work has significantly advanced the accountability and transparency of otherwise black-box machine learning algorithms, the stringent requirements of sensitive environments where AI is deployed necessitate a broader range of explanation techniques. In this section, we discuss some of the major limitations of current XAI methods. I also acknowledge that some research has already been conducted in these areas, and I do not claim to be the first to explore these ideas. However, I believe these topics are understudied and worthy of further investigation.

I argue that one of the greatest limitations of XAI is the focus on local explanations for providing interpretability, that is, an explanation for a single prediction [13]. The issue with such explanations is that they may be incredibly misleading when trying to calibrate our trust in a model [25]. The authors of LIME [14] demonstrate that, for instance, a network meant to distinguish huskies and wolves tended to focus on the distinction between indoor versus outdoor settings for huskies and

wolves respectively. While the implications for understanding the generalizability of a model are clear from this example, it relies on the user not only having queried LIME enough such that they could notice the trend but it also has the potential to ignore issues that the user failed to imagine during their testing. How would the network react to seeing husky or wolf puppies, for instance? Further, any insights gained would have to be interpreted by the user. What if, for example, it is not the outdoor setting that tricks the network into thinking that a husky is a wolf, but rather the presence of trees?

Local XAI is valuable, particularly in settings where individual decisions may require additional transparency for legal concerns [26, 27] or when such decisions may require action beyond classification, however this does not negate the need for global explanations, those which seek to provide greater interpretability for the model as a whole [13]. This aspect of explanation is understudied but explored in more detail here.

Additionally, XAI research predominantly focuses on supervised learning environments, with the expectation that these methods will transfer to unsupervised learning scenarios. However, this assumption may not always be valid. For instance, in supervised learning, multiple distinct classes each have unique characteristics. In contrast, anomaly detection operates under the premise of a single type of data—normal data—with some instances deviating from this norm. Here, the objective shifts from identifying what is unique about a specific instance or class to understanding how it *differs* from the normal data.

Finally, XAI tends to ground its explanations in the input space [14, 15, 17]. While this is a natural choice, with the rise of embedding structures [28] and in particular transformer embedding structures [29], the input to a neural network may no longer have the interpretable semantic meaning as deep embedding features may not be, on their own, disentangled [30]. Despite this, the need for explanation in such systems still exists, and therefore the need for XAI techniques that do not assume interpretability of the input space are required.

1.2. Limitations Of Existing Fair Machine Learning

Existing fairness research typically works by enforcing fairness with respect to individual dimensions of identity, or multiple dimensions of identity individually [31] under the theory that bias against particular groups may exist in the underlying data and that this bias can be exerted

through the interaction between the PSVs and the label space [6]. Despite machine learning models not typically having direct access to PSVs, a network can still pick up on the bias of the data and find surrogate values for PSVs [21, 32].

While this may address some forms of overt bias in machine learning, it does not align with the prevailing theories in the social sciences about how discrimination occurs, particularly concerning the theory of intersectionality which posits that dimensions of identity meaningfully intersect in ways different than the sum of their parts [33] e.g. the experience of a transgender man is not equivalent to the experience of being transgender plus the experience of being a man, but rather the intersections of these dimensions influence his overall experience. Some work has addressed these concerns [31, 34], though it remains understudied.

Further, I believe that the existing approach to fairness ignores the mathematical reality of the fairness problem. To assert that fairness is fundamentally a data problem ignores the reality that a network's decisions are based on an interaction between data and an algorithm. In the case of artificial neural networks, for instance, a network will generally converge around some local minimum with respect to the loss of the model, and that local minimum may hold bias towards (a) particular group(s). Decisions such as the architecture or other hyperparameters may influence where these local minima are, therefore how likely one is to arrive at a particular minima, and therefore which groups, if any, the algorithm may hold bias against.

Finally, the idea that labeling bias is the sole or main cause of unfairness is also unjustified, with empirical evidence suggesting that even unsupervised learners may behave unfairly with respect to certain groups [32]. Therefore, the mechanisms under which a network may become unfair in the absence of labeling bias require further investigation.

1.3. Our Contributions

To address the aforementioned limitations of XAI and fair machine learning, this dissertation makes the following contributions:

- We explore understudied areas of XAI, particularly with respect to:
  - Creating better global models of explanation (Chapters 4, 5)
  - Extending XAI beyond supervised learning tailored to the specific requirements of those types of learning (Chapters 3, 4)

TABLE 1.1. Relation between the explanation chapters & the limitations discussed in Section 1.1.

Chapter	Beyond Local Explanation	Beyond Supervised Learning	Not Reliant On Input-Space Interpretability	Machine-To-Machine Explanation
Cooperative Counterfactual-Based Knowledge Distillation: A Learner Agnostic Approach			✓	✓
Model Agnostic Relative Explanations for Anomaly Detection Using Diverse Counterfactuals		✓		✓
An Exemplars-Based Approach for Explainable Clustering: Complexity and Efficient Approximation Algorithms	✓	✓	✓	✓
Identification and Uses of Deep Learning Backbones via Pattern Mining	✓		✓	✓

  - Without the assumption of an interpretable input space (Chapters 2, 4, 5)

- We propose the novel concept of machine-to-machine explanation, and demonstrate that both new and existing XAI techniques can be used for tasks beyond interpretability to humans, particularly for additional computational tasks.
- We demonstrate that the existing approach to fair machine learning is fundamentally insufficient to handle how unfairness and discrimination are theorized to occur in the social sciences and mathematics, particularly with regard to:
  - Intersectionality (Chapters 6, 9)
  - Bias in data beyond labeling bias (Chapters 7, 8, 9)
  - Bias caused by the interaction between algorithm and data, rather than data alone (Chapters 7, 8, 9)
- We explore the intersection between explainability and fairness. Using our idea of machine-to-machine explanation, we demonstrate that machines can explain how they act unfairly and these explanations can be used to rectify such issues.

1.4. Summary Of The Dissertation

1.4.1. Cooperative Counterfactual Based Knowledge Distillation: A Learner Agnostic Approach. Here, we explore using an existing explanation approach, counterfactual explanations [35], for a task beyond interpretability to humans, specifically knowledge distillation. Counterfactual explanations are a paradigm in XAI that seeks to ask the question of how an instance would need to change in order for the prediction to be something different. For example, how would an image of a dog need to change for the model to believe it was a cat? If the features removed in the image are the most "doglike" features (eg snout, dog ears) and the features added are the most "catlike" (eg whiskers, cat ears), then this perhaps indicates that the model understands these concepts well and we can calibrate our trust in the model accordingly.

Rather than using this algorithm for a human interpretable explanation, we use this for the task of knowledge distillation - attempting to encode learned knowledge from one model and pass it to another. Here, the teacher uses a counterfactual optimization algorithm not to change the label of the instance, but to make it look *more* like the actual class label. Therefore, the teacher encodes learned information about the specifics of each class before passing the virtual instance to the student. In the chapter, we demonstrate that this introduces a focus mechanism where any model can act as either a student or teacher wherever appropriate and can be completed without sharing data.

This section functions as our purest version of machine-to-machine explanation and is the only technique to not introduce a new form of explanation. **A previous version of this paper is published in proceedings of AAAI24 [36]**

1.4.2. Model Agnositc Relative Explanations For Anomaly Detection Using Diverse Counterfactuals. To say that an instance is anomalous is to say that it differs from normality for some aspect(s). To explain why something is an anomaly is therefore a *contrastive* question - what is different about this instance from other instances? Further, since there may be many distinct ways to be considered normal, we argue that this explanation must be grounded *relative* to a set of nearby normal instances.

Here we develop a model-agnostic framework specifically tailored to the needs of explainable anomaly detection. We accomplish this by creating many different counterfactual explanations

for a single anomaly to turn it normal and mining distinct patterns among them and present the explanation as a small number of delta-vectors that, when added, would change the anomaly into a normal instance and, when subtracted, would turn nearby normal points into anomalies. We demonstrate that this technique works well for different types of anomalies, algorithms, and datasets.

This technique is not only used for explanation beyond supervised learning but it is also used for the machine-to-machine explanation task of self-supervised learning, in which the explanations are used to create virtual instances for a novel class.

1.4.3. An Exemplars-Based Approach for Explainable Clustering: Complexity and Efficient Approximation ALgorithms. Here, we develop a technique for simultaneous clustering and an explanation for that clustering. Borrowing the idea of exemplars from concept theory in psychology [37], we argue that an explanation can be crafted by selecting a series of important instances - the exemplars. We argue that this format of explanation is superior to naive exemplars (ie cluster centers) and demonstrates their practical utility. We prove that the simultaneous clustering and explanation problem is intractable but develop two approximation schemes, both of which create linear/logarithmic estimations of the original problem in polynomial time.

Further, many of our experiments rely on data that is not interpretable on its own, for example image and sentence embeddings. Here, LIME [14] style explanations would not be useful since the semantic meaning of the dimensions of the embedding is unknown. Every embedding, however, corresponds to some interpretable instance (eg the image/sentence which created that embedding). In this way, we are crafting a global explanation - an explanation for the entire clustering - and we do not rely on the interpretability of the feature space. Further, we demonstrate machine-to-machine explanation by showing that the exemplars can be used for instance transfer learning and outperforming simpler implementations. **A previous version of this paper is published in the proceedings of SIAM SDM24** [38].

1.4.4. Identification And Uses Of Deep Learning Backbones via Pattern Mining. In this work, we construct a framework for mining meaningful subgraphs out of a neural network's feed-forward hidden layers that activate for a particular user-defined concept but do not activate for others. Here, a concept is defined as any group of instances the user would like to explain as a

TABLE 1.2. Relation between the fairness chapters & the limitations discussed in Section 1.2.

Chapter	Handles Intersections	Beyond Label Bias	Towards Algorithmic Bias
An Empirical Investigation Of Intersectional Fairness	✓		
Causes Of Unfairness In Outlier Detection		✓	✓
Beyond Data Bias: A Proof Of Algorithmic Fairness Challenges		✓	✓
(Un)Fair Backbones In Neural Networks	✓	✓	✓

series of subgraphs in the network. We craft this problem as a tricriterion ILP, prove it intractable, and then develop a heuristic solution that exhibits Pareto optimality with respect to the three criteria.

Further, we demonstrate that these backbones can be used for many additional computational tasks. Most notably, we generate backbone explanations for how the model makes correct predictions and how it makes mistakes and develop a pipeline for rectifying these mistakes. In this way, we tackle the more complex XAI questions of "When do you succeed", "When do you fail", and "How can I correct an error?" [24].

This paper answers the above questions by creating a global explanation of the user-defined concepts. Because our explanation is not rooted in the input space, but rather in the activation space of the hidden units, this work may be of particular interest to those working with uninterpretable features. **A previous version of this paper is published in the proceedings of SIAM SDM24** [38].

1.4.5. The Intersectional Unfairness Paradox: An Empirical Investigation Of Intersectional Fairness. In Section 1.2, we argue that the main approach of fair machine learning - examining dimensions of identity individually - does not align with existing theories of intersectionality [33]. Here, we empirically investigate the consequences of this decision. We use multiple fair ML intervention algorithms and all eight datasets listed in a recent survey on the datasets used in fair machine learning [39] that contain two protected status variables. We create models fair with respect to either PSV or both PSVs individually and measure unfairness across three metrics

for both PSVs and the intersection of those PSVs. We demonstrate that even when models are made fair with respect to both PSVs, they are frequently made less fair with respect to intersections of PSVs, and often even less fair than the baseline of training without fairness interventions. These results indicate that the approach of optimizing fairness with respect to PSVs individually is fundamentally insufficient for creating equitable AI systems.

1.4.6. Beyond Data Bias: Proof of Algorithmic Fairness Challenges in Neural Networks. Fairness is often positioned as primarily a data problem - all data contains bias and an AI algorithm can exploit this bias to minimize loss [6]. However, this paper demonstrates that data alone cannot explain how unfairness occurs in models. Specifically, we prove that machine learning algorithms will develop their own biases outside that of the data influenced by factors such as model architecture and hyperparameter selection. We also demonstrate that the expected value for fairness can be calculated before training either exactly via an integral over parameter space or approximated by a dimensionality reduction over parameter space.

1.4.7. Foundations of Unfairness in Anomaly Detection - Case Studies in Facial Imaging Data. Here we seek to provide a model of how unfairness occurs in the absence of label bias, specifically for anomaly detection. This model is based on four factors: incompressibility, sample size bias, label attribute noise, and spurious feature variance. We demonstrate through hypothesis testing that no single cause of unfairness is sufficient to explain the phenomenon alone, the combination of all factors is sufficient, and that no factors are redundant. This work serves as a model for how unfairness can occur in these settings, and future work can deal with how to overcome it.

1.4.8. (Un)Fair Backbones In Neural Networks. Here we extend our work on backbone explanations to the fairness domain. As discussed in the description for the backbone section, a backbone can be created to explain any concepts the user would like so long as they can find a group of instances that define that concept. In this case, we create backbones for the concepts of fair vs unfair actions and explore several uses for these backbones. Specifically, we examine a post-hoc method of decision adjustment, a post-hoc zero-shot learning approach to maintain the majority of performance while reducing unfairness, and an in-situ focused dropout approach to train fairer models.

CHAPTER 2

Cooperative Knowledge Distillation: A Learner Agnostic Approach

Abstract

Knowledge distillation is a simple but powerful way to transfer knowledge between a teacher model to a student model. Existing work suffers from at least one of the following key limitations in terms of direction and scope of transfer which restrict its use: all knowledge is transferred from teacher to student regardless of whether or not that knowledge is useful, the student is the only one learning in this exchange, and typically distillation transfers knowledge only from a single teacher to a single student. We formulate a novel form of knowledge distillation in which many models can act as both students and teachers which we call cooperative distillation. The models cooperate as follows: a model (the student) identifies specific deficiencies in its performance and searches for another model (the teacher) that encodes learned knowledge into instructional virtual instances via counterfactual instance generation. Because different models may have different strengths and weaknesses, all models can act as either students or teachers (cooperation) when appropriate and only distill knowledge in areas specific to their strengths (focus). Since counterfactuals as a paradigm are not tied to any specific algorithm, we can use this method to distill knowledge between learners of different architectures, algorithms, and even feature spaces. We demonstrate that our approach not only outperforms baselines such as transfer learning, self-supervised learning, and multiple knowledge distillation algorithms on several datasets but it can also be used in settings where the aforementioned techniques cannot.

2.1. Introduction

NOTE: A previous version of this paper is published in AAAI24 [36, 40] with co-authors Ian Davidson and Stephen Wong

Knowledge distillation is a simple and elegant approach that allows one machine (the teacher) to instruct another machine (the student). Typically, the teacher model is more complex than the

student model, and knowledge distillation compresses models for efficiency [41], though more recent work explores improving performance as well [42]. However, existing knowledge distillation has its limitations. First, offline knowledge distillation, that is, a trained teacher teaching an untrained student, assumes that all of the teacher's knowledge is good and should be learned by the student even if the teacher performs worse than the student. Second, it is unidirectional and singular; one teacher informs one student, and students do not inform teachers.

In this work, we extend knowledge distillation to novel settings by creating what we call cooperative distillation. This is useful in domains where there are multiple learners, each of which can be considered a semi-expert deficient in one or more particular aspect(s) of a task, and can help overcome each other's limitations. This setting is not covered by existing distillation work. Consider our FashionMNIST dataset experiment. Here, we create ten classifiers (one for each class) trained with one class being undersampled by 95% to induce a conceptual deficiency. A model might understand the majority of clothes it sees, but since it hasn't seen many, say, ankle boots, it struggles to classify them correctly and will rely on other models to teach it this concept. This will require targeted and multidirectional transfer: this model needs to be taught only about ankle boots and can be a teacher for other classes.

In the tradition of knowledge distillation simplicity, we propose a learner-agnostic, counterfactual-based cooperative approach. Consider an instance x which model i can predict correctly, but model j cannot. We say that model i is a qualified teacher to model j for the specific instance x. Our method will have model i teach model j about x by generating a new type of quintessential counterfactual x' which can be added to j's training set. We call this type of counterfactual quintessential because instead of modifying the instance to change its label, we have the model i make this instance look even more like the true class. Counterfactuals were chosen as the method to generate virtual instances since they are both model agnostic and virtual instance generation is driven by the model. Our approach is multidirectional as any model can teach any other and focused as we transfer only some instances between models via counterfactuals.

Our work can be viewed as being in a similar setting to domain adaptation and transfer learning but has notable differences. Typically, domain adaptation is from a chosen single expert source to a single novice target, whereas our work is cooperative between semi-experts with no need to

choose a target/source. Further in our work, the domain of the teacher and student models are the same which is not the case for transfer learning. Our contributions are:

- *New Style of Distillation.* We propose a simple yet powerful approach to a new form of distillation we call cooperative distillation. This is achieved using a novel type (quintessential) and use of counterfactuals.
- *Robust Across Learners.* Experimental results are promising for a variety of basic (i.e., decision trees) and complex learners (i.e., convolutional neural networks) (see Experimental Section, particularly Table 2.2).
- *Robust Across Settings.* We demonstrate our method's good performance under various settings, including distilling between different architectures/algorithms, high-performance models, low-performance models, mixtures of high and low-performance models, and varying degrees of feature overlap.
- *Outperforms Baselines.* Our approach can significantly outperform multiple state-of-the-art and state-of-the-practice baselines in transfer learning, self-supervised learning, and knowledge distillation. (see Table 2.2 which summarizes all our experiments).

We begin this paper by outlining related work and describing our approach. We then provide experimental results for various learners, followed by a discussion on our method's strengths and weaknesses including our hypotheses about why our method works, after which we conclude.

2.2. Related Work

The field of knowledge distillation exists to transfer learned information from one learner to another, typically a more costly high-performance model to a lightweight model [41] [43] [44] in the same task. This is distinct from transfer learning which, by definition, uses a learner in a related but different source domain to assist in the training of a learner in the target domain [45]. Our work is further differentiated by distilling knowledge between semi-experts in a multidirectional fashion, as opposed to an expert to a novice.

Knowledge distillation literature can be categorized by two main factors: what is considered knowledge and the distillation scheme [44]. We first discuss how these questions have been answered by previous work and then present our novel knowledge paradigm.

TABLE 2.1. An overview of the knowledge distillation paradigms where L is the loss function, $\mathcal{L}$ is a function to calculate the differences between the teacher (t) and student (s), and Φ and Ψ are comparison functions between different activations and relations, respectively.

Name	Knowledge	Loss	Corresponding Works
Response Distillation	The teacher annotates instances using its logits for the student	$L(z_t, z_s) = \mathcal{L}(z_t, z_s)$	[41, 42]
Feature Distillation	Student is trained to replicate teacher's hidden layer activations for training instances	$L(f_t(x), f_s(x)) = \mathcal{L}(\Phi_t(f_t(x)), \Phi_s(f_s(x)))$	[46, 47, 48]
Relation Distillation	Students are trained to have similar relations between multiple aspects of the model compared to the teacher	$L(f_t, f_s) = \mathcal{L}(\Psi_t(f_{t_i}, f_{t_j}), \Psi_s(f_{s_i}, f_{s_j}))$	[49, 50]
Counterfactual Distillation	Training data is encoded with information from the teacher to teach the student about the class	$L(X \cup X') = L(X) + L(X')$	Ours

Knowledge Distillation Paradigms. There are three general categories of knowledge distillation algorithms: response/output distillation in which a student learns to replicate the output of the learner by calculating loss between the student's logits and those of the teacher [41] [51], feature distillation which trains a student to mimic the teacher's parameters, such as hidden layer weights and biases, [46] [48] [52], and relation distillation, which is concerned with the relations between multiple parts of the model such as multiple feature maps [53] [50], feature maps and logits [49], or pairwise similarities between the input data and output distribution [54].

Distillation schemes are also important to categorize the different forms of knowledge distillation. Knowledge can be distilled from a learned teacher model to a student in offline knowledge distillation [41] [43] [50], or while the learned model is being trained in online distillation [55] [56]. It is important to note that some of these approaches do consider multiple learners both students and teachers [57] (one of our contributions), however, the task for those approaches is to distill knowledge during the lurking process, whereas we are distilling knowledge between trained models, which is novel for this setting.

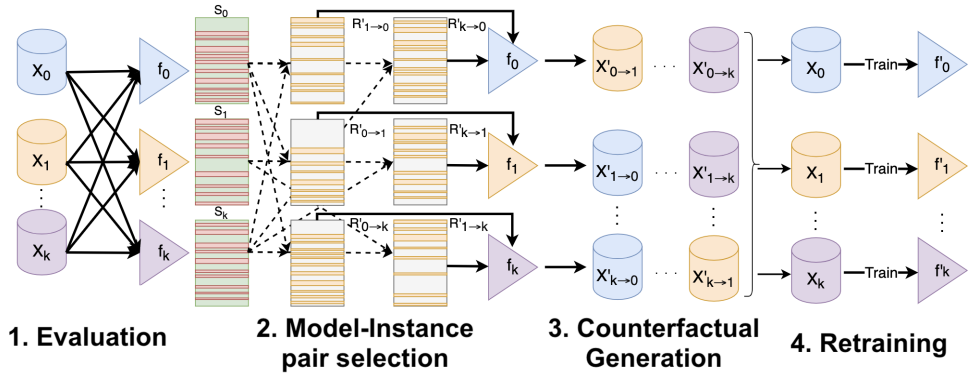


FIGURE 2.1. Pipeline for our method. Each model and its corresponding datasets are color-coded. Given $k+1$ models f_0 through f_k and datasets X_0 through X_k, we find which instances among the training sets each model can correctly predict (rows colored in green), creating a set of indices for each model (step 1). For all permutations of groups of two of these sets (S_i, S_j), we find $R_{i \to j} = S_i - S_j$: instances for which model i is a teacher for model j (rows highlighted in yellow, step 2). We then create counterfactuals using the appropriate teacher models and instances, labeled $X'_{teacher \to student}$ (step 3), shuffle the new, augmented instances into the training sets of each group, and retrain the models to create augmented models f'_0 to f'_k (step 4).

A New Use for Counterfactuals - Cooperative Distillation Rather than encoding knowledge into output logits, parameters, or relationships, our work embeds learned information in the data itself by creating virtual instances (counterfactuals) and passing them on to the training sets of other models. Further, it should be noted that the distillation scheme is also a special case of offline knowledge distillation, as instead of a student learning from a teacher, each model will act as both teacher and student simultaneously, something not explored in offline knowledge distillation. This is distinct from both self-distillation [58], in which a single model acts as both teacher and student, as our method uses multiple models, and online distillation [55] in which models distill knowledge during training, as our method leverages trained models.

2.3. Our Approach: Cooperative Distillation

Our method is a form of offline knowledge distillation but with two important enhancements. First, it considers distillation across multiple models where each model can act as both a teacher and student, rather than distilling from a single teacher to a student. Our second innovation uses counterfactuals to generate targeted instances to transfer rather than distilling knowledge across all instances as in traditional knowledge distillation. This is a form of cooperation as the

student identifies instances it performs poorly on and the teacher creates an easier-to-understand counterfactual.

Our approach takes three fundamental steps:

(1) **Expertise Identification:** Model i selects instances (I) it can accurately predict.

(2) **Deficiency Identification:** From I, every other model j finds instances it cannot predict $R_{i \to j} \subset I$.

(3) **Cooperative Distillation:** For each instance $x \in R_{i \to j}$, i creates counterfactual x' to be added to j's training set.

2.3.1. Expertise and Deficiency Identification. Since each model may have limited knowledge of the domain, it is crucial that models acting as teachers only do so in settings where they are "qualified" teachers. A model i is considered qualified to teach a student model j about an instance x if and only if model i correctly predicts instance x and model j does not. In this way, students are only taught concepts they fail to understand and only from qualified teachers.

To decide which models act as students and which act as teachers for different instances, we first pass all of the training data X (this can be done without sharing data, see Figure 2.2) to each of the models and collect sets of indices of the instances that model can predict correctly. Let S_i be the set of instance indices correctly predicted by model i. Let $R_{i \to j}$ be the set of instance indices that model i correctly predicts that j does not. Formally:

$$R_{i \to j} = S_i - S_j \tag{2.1}$$

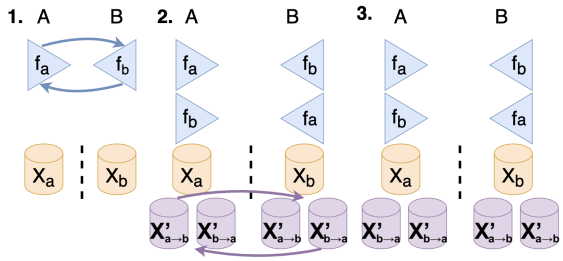


FIGURE 2.2. Non-Data Sharing Scenario: Our approach for deploying the approach while maintaining data privacy. Institutions may share models, but not data. Models are exchanged (step 1), our technique is applied to create a subset of the virtual instances (step 2), those virtual instances are shared, and the models are retrained (step 3).

This can be accomplished even if datasets cannot be shared. Consider two groups/organizations/sites who can share models, but not data. After sharing models, they can use our approach to generate virtual instances on their respective datasets and only share those virtual instances. This process can be visualized in Figure 2.2. Equation 2.1 must be computed for every permutation of two models. Therefore for k models, the complexity of this subroutine is $O(P_2^k|X|)$, where $|X|$ is the size of the training data.

2.3.2. Quintessential Counterfactual Generation. Counterfactual algorithms generate a virtual instance x' given three pieces of information: the model f, an instance x, and desired output y' such that x' is similar to x and $f(x') = y'$ [59]. Most work creates contrastive counterfactuals which "flip" the label, $(f(x) \neq f(x'))$ whereas our method generates quintessential counterfactuals - those which the existing prediction is made greater.

The instance selection mechanism previously described finds the appropriate instance-teacher pair (f, x). We chose to set $y' = f_i(x) + \alpha(y - f_i(x))$, where α encodes the teacher model's influence. The closer α is to 1 the closer y' is to y (the true class label), and the closer α is to 0, the closer y' is to $f_i(x)$. This allows the teacher model to inject knowledge about the class into the instances. All experiments in this paper set α to 0.5, as this is an even balance between the original instance and a theoretical instance of perfect certainty. Conceptually, our counterfactual generation process can be visualized in Figure 2.3.

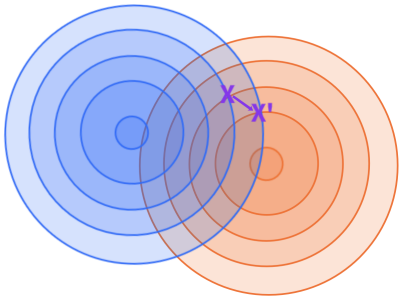


FIGURE 2.3. Quintessential counterfactual generation illustrative example. Each model's decision surface is a contour map with each circle representing 20% confidence of the correct class. The orange model (teacher) predicts instance x with 60% confidence, and the blue model (student) mispredicts x as its confidence in the correct class is only 40%. The teacher model creates a virtual instance x' it believes is more typical of the class.

The counterfactuals are assigned the correct label for the original instance (y), the augmented instances are added to the training set of the student, and the model is retrained. This process is described in Figure 2.1 and Algorithm 1. Counterfactuals for model j from model i are generated as below:

$$\forall x \in R_{i \to j}\ argmin_{x'}\ d(x', x) + \lambda |f_j(x') - y'|^2 \tag{2.2}$$

where $R_{i \to j}$ are the instances model i can teach model j.

Here, d is a distance metric, in our case Manhattan distance, and λ is a balance term. As is standard [12], we set λ to the maximum value for which a solution will converge. For models with differentiable parameters, such as neural networks, we use gradient descent via Adam, and for models without differentiable parameters, the non-gradient-based particle swarm optimization [60]. The complexity of generating a counterfactual is constant, making the cost $O(k|X|)$ for k models.

Assuming the cost to train k models is proportional to the data size, the algorithm's time complexity is $O(P_2^k|X| + k|X|)$, or with a constant number of learners $O(|X|)$.

Algorithm 1 Cooperative knowledge distillation

Require: $k + 1$ trained models $F = \{f_0, f_1, ..., f_k\}$, respective datasets $D = (X_0, Y_0), (X_1, Y_1), ..., (X_k, Y_k)$ and balance term α.
Ensure: k retrained models

```
S := new list of sets
for all (X_i, y_i) ∈ D do
    for all f ∈ F do
        S_i = f.scorePredictions(X_i, y_i)
AugmentedInstances := new list of instances
for all (S_i, S_j) ∈ S do
    R_i := S_i − S_j (Eq. 2.1)
    for all (x, y) ∈ D[R_i] do
        y' := f_i(x) + α * (y − f_i(x))
        cf := GenerateCFs(f_i, x, y') (Eq. 2.2)
        AugmentedInstances[j].append(cf)
for i = 0 to k do
    newDataset := AugmentedInstances[i] + D_i
    f_i.train(newDataset)
```

2.3.3. Extensions for Mismatched Feature Sets. Some practical situations exist where the feature sets are not identical but overlap. This can occur for a variety of reasons including

			Transfer Learning	*SSL*		*Knowledge*	*Distillation*		*Data-Pollution*
Approach\Experiment	**Baseline**	**Ours**	**Parameter Transfer**	**GAN**	**Mixup**	**Response-Based KD**	**KD as Pretraining**	**Parameter Based KD**	**Add Training Data Together**
Exp. 1 CL MLP	60.98%	**68.68%**	N/A	65.13%	52.56%	59.29%	52.43%	61.24%	67.01%
Exp. 1 AE MLP	86.04%	**87.74%**	N/A	86.58%	62.71%	60.00%	85.87%	N/A	84.43%
Exp. 2 CL 1 D-Tree	63.41%	**67.58%**	N/A	62.41%	55.63%	58.99%	N/A	62.53%	69.29%
Exp. 2 AE MLP	86.04%	**87.32%**	N/A	86.58%	62.71%	58.03%	86.72%	N/A	84.43%
Exp. 3 Model 1 D-Tree	56.71%	**57.44%**	N/A	56.38%	53.38%	54.55%	N/A	N/A	60.72%
Exp. 3 Model 2 D-Tree	43.45%	**62.36%**	N/A	51.49%	57.16%	55.45%	N/A	N/A	62.54%
Exp. 3 Model 3 D-Tree	53.47%	**63.04%**	N/A	55.45%	58.22%	62.27%	N/A	N/A	55.17%
Exp. 4 Model 1 D-Tree	56.71%	**66.24%**	N/A	56.38%	60.89%	54.55%	N/A	N/A	60.72%
Exp. 4 Model 2 NB	62.37%	**77.56%**	N/A	64.69%	70.41%	54.13%	N/A	N/A	64.03%
Exp. 4 Model 3 SVM	54.45%	**59.08%**	N/A	55.78%	52.96%	54.13%	N/A	N/A	58.42%
Exp. 5 Median CNN	76%	83%	82%	81%	79%	73%	**86%**	80%	86%

TABLE 2.2. Median results from 10 to 90 experiments. Methods that cannot be used for a particular dataset are marked with N/A. In all four of the main experiments (1-4) our method outperforms all baselines and competitors. The stress test in Experiment 5 designed to test our method's ability to handle many models achieves the second highest performance, with knowledge distillation as pretraining performing best. The baselines are models are trained without any augmentation.

if data is collected from different locations or sites. All models are tested on the same test set, which also comes from a different dataset from another site containing no instances from any training or validation sets. Consequently, some datasets may have different features, and we must therefore pass instances into training sets of incompatible feature spaces. To deal with this, we normalize continuous ratio and interval data between zero and one, and one-hot encode categorical and discrete interval data, setting missing features to zero, as suggested in [61] [62].

2.4. Experiments

To demonstrate our claims we conduct six experiments, using models generated from five different algorithms trained on nine datasets for four tasks. Recall that we claimed that our model agnostic cooperative distillation involves focused and multi-way distillation. Each experiment is meant to examine one particular aspect of these claims and compare them to other relevant state-of-the-art and state-of-the-practice methods. We next discuss the implications of each experiment and provide detail in subsequent subsections.[1]

[1]To aid in reproducibility, code is provided on GitHub https://github.com/MLivanos/Cooperative-Knowledge-Distillation

- Experiments 1 and 2 examine how our approach handles distilling between different architectures (Experiment 1) and algorithms (Experiment 2), as well as differing amounts of data and performance. This asymmetrical setting leads to different numbers of counterfactuals generated for each model (see Figure 2.6).
- Experiments 3 and 4 use three models to test our model's multidirectional claim as each model has a small amount of training data and all need to cooperate to master the domain. Further, these models start at a relatively weak performance, meaning we are also testing our focus mechanism to ensure that only relevant knowledge is distilled. Experiment 4 pushes the limits on our model-agnostic claim as we have a decision tree, Naive Bayes classifier, and SVM cooperating via our method.
- Experiment 5 tests many aspects of our claims at once: the ability to distill between many semi-expert models doing focused transfer. We create a situation where the training data is made deficient in exactly one class for each of the ten convolutional neural networks.
- Our last experiment addresses our claim that our method can be used in settings with different amounts of overlapping features by starting with perfect feature overlap and iteratively removing features to test the correlation between feature overlap and accuracy increase. See Figure 2.7.
- Notably, with the exception of Experiment 5, no two datasets have a single instance in common with each other, instead relying on the process outlined in Figure 2.2 to accomplish our technique without sharing data.

The results of our experiments are summarized in Table 2.2. An important aspect of the results is that our method improves all 20 models trained, which did not occur with any competitor. The rest of this section will discuss the results and implications of each experiment individually.

All models discussed trained to convergence, and hyperparameter selection maximized validation set accuracy.

Baselines and Competitors. Each experiment tests against several competitors: parameter transfer [63], self-supervised learning techniques including generative adversarial networks (Deep Convolutional GAN (DCGAN) [64] for image datasets and TabGan [65] for tabular datasets) and mixup [66], and knowledge distillation, including response-based offline knowledge distillation [41], response-based knowledge distillation which achieved state-of-the-art accuracy on the Imagenet

dataset [42], and finally a recent, state-of-the-art feature-based knowledge distillation [47] algorithm. We also compare against a baseline of the original model's accuracy (without distillation) and an idealized setting in which all training data is combined. This last setting may be unrealistic due to data proprietary, privacy, or availability and is thus compared separately.

Experiments 1 & 2: Cross-Architecture/Algorithm Distillation. These experiments use three different datasets to predict if a used car is expensive (>\$20,000) or inexpensive (≤\$20,000). Each dataset comes from a different website curated between 2020 and 2021. Datasets 1 and 2 come from Craigslist [67] and Auction Export [68] respectively, and are used for training and validating models. A test set from Car Guru [69] simulates a future distribution all models will have to predict. We expect that each data set covers different types of cars (eg makes, models, years) in different depths.

Experiment 1 examines how our technique can distill knowledge between models of different architectures that were tuned for the different data sets. The Craigslist (CL) and Auction Export (AucEx) models use neural networks of different architectures: the former with one hidden layer with 512 neurons, the latter with one hidden layer with 1024 neurons, both with leaky ReLU activation functions for hidden layers and sigmoid for the output layer. These architectures create models with test set accuracies of 60.98% and 86.04% for the CL and AucEx models respectively, and our method improves this to 68.68% and 87.74%. These results not only demonstrate a boost to both model's performances but also show that a low-performance model can teach an high-performance model - a result that no other distillation technique could replicate. In the case of the AucEx model, our method performed even better than the idealized case of training using all available instances. A total of 18923 instances were distilled to the CL model and 1613 to AucEx.

In Experiment 2, the AucEx model is the same neural network, and the CL model is now a decision tree (minimum samples leaf set to 20) to explore how well knowledge can be distilled between different algorithms. The AucEx model's baseline performance is identical to the above experiment, and the CL model achieves baseline test accuracy of 63.41%. Our method successfully elevates the performance of the models to 67.01% and 87.32% for the CL and AucEx models, respectively, again surpassing all competitors. A total of 30842 instances were distilled to the CL model and 1817 to the AucEx model.

Experiment 3 & 4: Small Data Distillation. Experiment 3 tests how well knowledge can be distilled between three low-performance models built from small datasets. Four datasets are used, each predicting the presence or absence of heart disease from hospitals at different locations: Long Beach (Model 1), Switzerland (Model 2), Hungary (Model 3), and Cleveland [70], all sourced from [71]. The Cleveland was chosen as the test set since it contained all of the features of the previous three, making the evaluation fairer. We remove features from each dataset individually if at least 25% of instances are not reported and train decision trees for each dataset. Baseline test set accuracies for each model are 56.71%, 43.45%, and 53.47%, which are improved to 60.72%, 62.54%, and 55.17%, respectively by our method, greater than all applicable competitors, and in the third case, beating the idealized scenario of adding all instances. This demonstrates our method's ability to distill knowledge even between low-performance learners.

Experiment 4 uses the same datasets, however instead of using models from the same algorithm, knowledge will be distilled between several different algorithms: a decision tree, naive Bayes, and support vector machine classifier. Model 1 is the same decision tree as in experiment 3, and models 2 and 3's baseline performances stand at 62.37%, and 54.45%, respectively. With comparably stronger baselines, our method elevates performance to 66.24%, 77.56%, and 59.08%, surpassing all competitors and the idealized scenario of pooling together all instances. This experiment provides further evidence to suggest that our method not only can handle distilling between different algorithms but is largely invariant to algorithm choice and that stronger teacher models tend to help more than weaker ones.

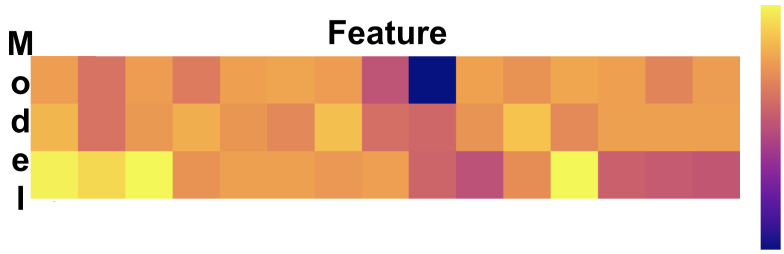
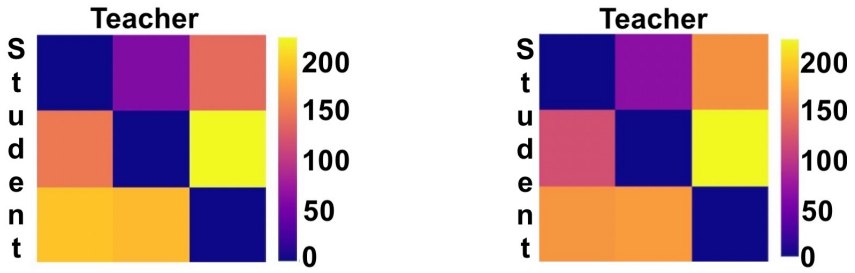


FIGURE 2.4. Heatmap for the knowledge distilled into different models of Experiment 4. Columns are features, and the rows represent the average change (yellow is positive, blue is negative) to move an instance to the diseased class. For example, the bottom left tile shows an increase in age (first column) is associated with heart disease whilst the middle top row indicates a reduction in ST-Depression (ninth row) decreases heart disease.



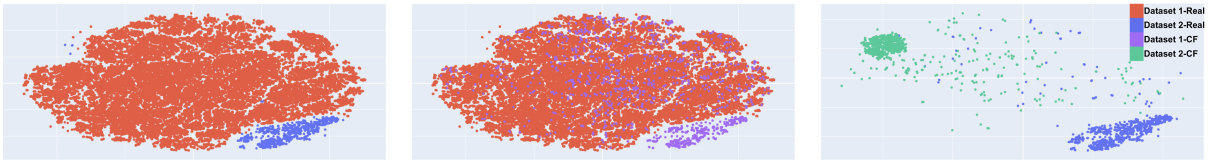
(a) Experiment 3 (b) Experiment 4

FIGURE 2.5. Heatmap to visualize the number of counterfactuals distilled to each model. Rows are students and columns teachers, with yellow implying more instances and blue less.

Experiment 5: Many Model Distillation Between Semi-Experts. We create ten datasets from the grayscale image dataset FashionMNIST [72], each of which is undersampled (by 95%) in one particular (and different) class. This creates a scenario in which all models are deficient at predicting a particular class but other models are proficient in that class. This is a rigorous test of our claims of multi-way and selective distillation. To produce higher quality counterfactuals, we optimize images only over the 50% most variable pixels of their class.

The median baseline accuracy for the ten models rests at 76%. Since each model acts as a teacher to the other models, each model would receive thousands of new counterfactuals for the under-represented class, resulting in redundant counterfactuals that elevate performance to 79%. After removing similar counterfactuals via geometric set-cover, we improve median accuracy of 83%, approaching our topline (no undersampling) accuracy of 86%. Since all models are networks of the same architecture, we could apply a greater range of competitors such as parameter transfer. This is the only experiment in which one of the competitors (knowledge distillation pretraining) surpasses our technique.

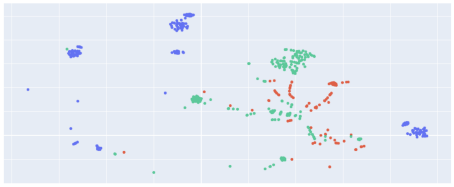
Experiment 6: Sensitivity to Feature Overlap. Three random and non-overlapping subsets are extracted from the Statlog German Credit dataset with 400, 400, and 200 instances. We generate two models from the larger subsets and test on the third. Since we are using the same dataset, there is a perfect overlap between the features. Iteratively, we remove different features at random from both datasets until they only have two in common. This sensitivity test examines the effect of feature overlap on our method's performance and is repeated five times due to the random



(a) Distribution of original datasets for datasets 1 and 2

(b) Original and augmented instances for dataset 1

(c) Original and augmented instances for dataset 2



(d) Distribution of original datasets

(e) Distribution of original datasets and added augmented instances

FIGURE 2.6. All data points, original and augmented, for both datasets of Experiment 1 (top; subfigures a-c) and Experiment 4 (bottom; subfigures d,e), projected into two dimensions using t-SNE. Each model creates instances similar to their own data while being distinct from the original data points.

nature of feature removal. Since the datasets are random samples from the same distribution, performance increase is small compared to other algorithms but largely invariant to feature overlap, with correlation coefficients of $-9.45 * 10^{-4}$ and $1.17 * 10^{-3}$ for each model (Figure 2.7), averaging a very small ($2.25 * 10^{-4}$) correlation between increase in accuracy and difference in feature space. We speculate that perhaps this is due to the data distributions becoming more diverse as features are removed, allowing for more distillation between the models. This indicates that data similarity may be an important indicator of success, and differences between datasets can overcome the feature overlap problem.

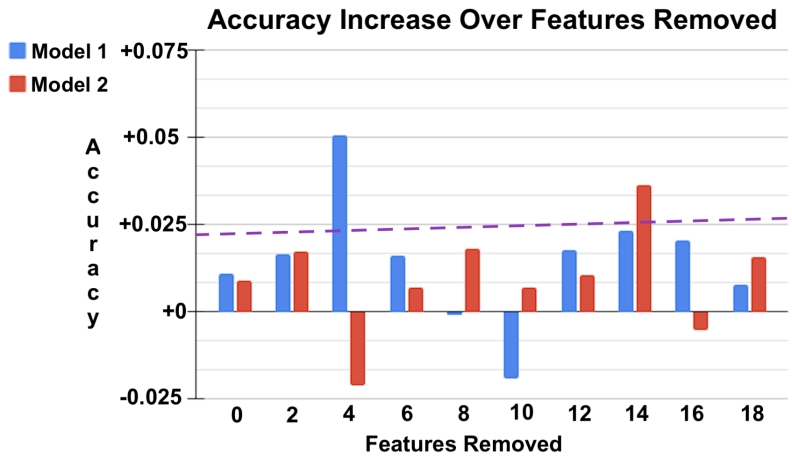


FIGURE 2.7. Improvement in accuracy over the number of features removed (feature overlap becomes smaller along the X-axis). There is no significant correlation between feature overlap and performance gain, as demonstrated by the trendlines which average a slope of $2.25 * 10^{-4}$.

2.5. Understanding The Mechanisms of Distillation

We test two hypotheses to explain our previous successful experiments. First, we believe our method introduces novel virtual instances to a learners' dataset increasing diversity and allowing better generalization. To test this, we visualize the counterfactuals our method creates using t-SNE (Figure 2.6) and find that counterfactuals, while being distinct from original data, create a similar distribution to it.

Our second hypothesis is that the teacher encodes information it believes to be important for classification into each instance. To test this hypothesis, we examine the average modification to original instances and separate them into counterfactuals of the positive (δ_+) and negative (δ_-) classes. Subtracting δ_- from δ_+ provides a vector that details the changes made to features to move them from one class to another. Figure 2.4 demonstrates such knowledge distilled for Experiment 4. Here, models receive different information learned from their teachers. For instance, models 2 and 3 are provided with the information that someone is more at risk for heart disease given an increase in age, while the counterfactuals generated for model 1 produce relatively small changes to age. The diversity of these vectors illustrates how different models have discovered different patterns which may explain the models' performance increase.

The datasets of Experiment 2 have hundreds of features, making a visualization such as Figure 2.4 uninterpretable. Instead, how features were to move the output closer to the 'expensive' class. Features that decreased the most were mileage (by an average of 3660 miles), years prior to 2005, and models such as Prius, Outback, and Range Rover Sport, and makes Toyota and Smart. Counterfactuals had the strongest positive association with years above 2013, models F-450 Super Duty, GX, and LX 570, and makes Maserati and Porsche. The CL dataset had few Toyota listings with no exposure to Porsche or LX 570 vehicles, indicating that the AucEx model successfully introduced such instances to the Cl model, partially explaining the performance increase.

2.6. Discussion & Conclusion

Our experiments demonstrate our approach can distill knowledge between two or more models regardless of architecture, algorithm, feature overlap, and under small or large data settings. Since our method targets specific weaknesses of each model, we can distill knowledge between any combination of high and/or low-performance models, compared to traditional knowledge distillation techniques which tend to only distill knowledge from a single high-performance model to a low-performance model [41] [44]. Though our method performed well on real-world data sets it does have some assumptions. It assumes there is some overlap between the features of the data sets and most importantly, our method works best when the distribution of the datasets used to train models are significantly different from each other. Further, our method is fundamentally limited by the strength of counterfactual generation. Counterfactual explanations are easy to compute on tabular data but their performance on more complex data, such as images is more challenging. However, more recent approaches have found success in more basic image networks [73] [74], so as research progresses, we believe this limitation will be removed.

We show in Figure 2.5 the number of instances each model teaches to the others. Interestingly, this quantity is asymmetrical which will motivate future work to better understand the mechanisms of how each model teaches the others.

Conclusion We present a novel form of knowledge distillation that can be used between multiple models, in multiple directions and is focused. Each model simultaneously acts as teacher and student, distilling knowledge to the other by encoding learned information into virtual counterfactual

instances and passing them into the training sets of other models. Unlike other knowledge distillation algorithms, which always distill knowledge from the teacher to student, we use a targeting mechanism to ensure that teachers only distill correct knowledge tailored to a student's deficiencies.

In our four main experiments, our method beats the competitors studied, including state-of-the-art knowledge distillation algorithms. In a stress test to determine if knowledge could be distilled between many (10) models, our model surpasses all but one competitor and remains competitive. We find our method particularly useful in the setting where models can be freely shared, but raw data cannot, and the data sets share some features. This is common in medical imaging or finance communities where data is confidential. Given our method's strong performance on experiments simulating the aforementioned setting, we believe this to be a viable approach to knowledge distillation under such circumstances.

CHAPTER 3

Model Agnostic Relative Explanations for Anomaly Detection Using Diverse Counterfactuals

Abstract

Anomaly detection (AD) algorithms are frequently deployed in challenging environments in which the algorithm must identify instances for investigation, policing, and scrutiny. The high-stakes nature of applications such as fraud detection and the sensitivity of deploying AD on humans means transparency into how decisions are made is paramount. However, most AD algorithms can only identify which instances are anomalous and offer nothing about why these particular instances are anomalous. This is true not just for deep AD algorithms but even those non-deep methods particularly those that are property (i.e. density) based. In this paper, we generate a new style of explanation we call relative explanations which is well suited for AD. Rather than explaining why an instance is an anomaly by looking at solely its properties, we instead generate counterfactuals from the anomaly to explain why it is not a normal instance. To our knowledge, our work is the first work on model agnostic, task agnostic contrastive relative explanations for AD. We show this leads to both additive and subtractive explanations which is important for high-impact applications where shallower explanations would be insufficient.

3.1. Introduction

Anomaly detection (AD) is perhaps the most controversial of data analytic tasks given its typical purpose of identifying entities for investigation, policing, and scrutiny. Given a set of points, a subset of them are identified as anomalous and further investigated. Hence generating an explanation for AD algorithms is paramount if it is to be used on a wider variety of problems particularly those where the entities are people. Applications of AD in knowledge management view the anomalies as both noise (such as its use to remove hate speech) and signal (such as its application to target individuals for further investigation).

Explainable artificial intelligence (XAI) exists to provide human-interpretable justification for a machine's decision, however, most XAI work has been developed for supervised learning and clustering tasks and is not specifically designed to address the challenges of AD. Here, we propose a novel framework for explainable anomaly detection (XAD) that uses nearby normal points to generate relative explanations.

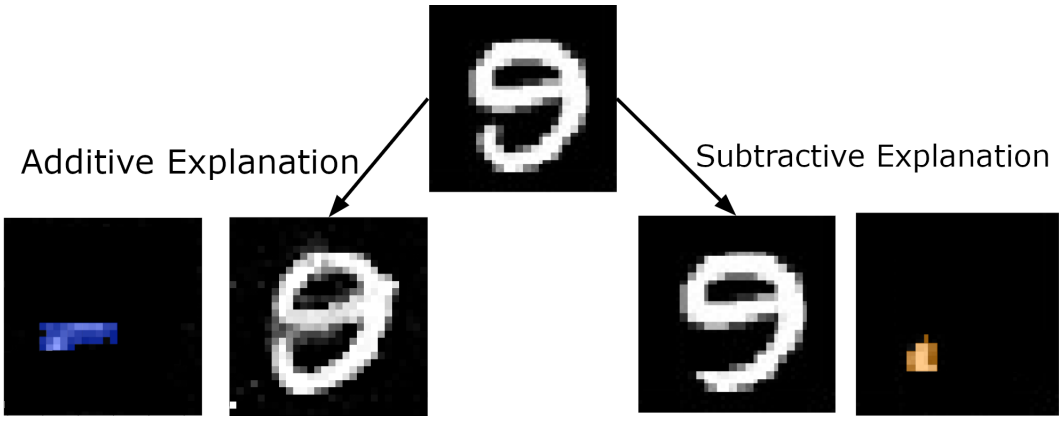


FIGURE 3.1. Experiment with MNIST dataset. Our method discovers that the anomalous numeral 9 (top) can be made normal by an additive explanation (bottom left) which completes the loop to make an eight. Alternatively, a subtractive explanation (bottom right) removes the elongated tail to make a more traditional nine. The addition/subtraction vectors can be visualized on the outside of the resultant image: blue indicates the addition of pixels and orange the subtraction.

The Need For AD Specific Explanations. Supervised learning XAI methods inherently ask the question, "Why does this instance belong to this class?". Methods like LIME [14], Anchor [75], and Shapley [76] produce attribution vectors highlighting the features that indicate inclusion in the class. In contrast, AD finds a group of normal points, and the anomalies are those that do not belong. Hence the key XAI question is "Why does the anomaly not belong to the normal group?". For this reason, we advocate for the use of constrastive relative explanations. Rather than trying to compare an instance to the learner's representation of normality as a whole, an explanation must be grounded relative to some nearby neighborhood(s) of normal points, and we highlight in what way(s) the anomaly contrasts with this nearby normal points.

Consider the illustrative example of the anomalous numeral in Figure 3.1 from the MNIST handwritten digits dataset [77]. One reason that this is anomalous is because, relative to a nearby neighborhood of nines, the extended loop at the tail is strange. Therefore subtracting/removing the end of the loop (highlighted in orange) will make it normal. However, this instance is also

anomalous relative to a nearby cluster of eights, and the contrasting feature of this neighborhood is the lack of a closed tail (highlighted in blue).

We argue that contrastive relative explanations are the most appropriate form of explanation because they address the underlying assumptions of anomaly detection and allow for multiple distinct explanations, all of which are necessary for a complete understanding of why the instance is anomalous.

The Need For Model Agnostic Explanations. A unique aspect of AD is the plethora of fundamentally different algorithms. Techniques such as local outlier factor (LOF) [78] even do not directly optimize a function like most learning tasks but rather look for data properties and hence are model-less. Deep learning methods such as autoencoder-based (AD) assume that instances that have high reconstruction error are likely anomalous. Because of the diversity in assumptions and underlying techniques used in AD algorithms, we focus on a model-agnostic framework.

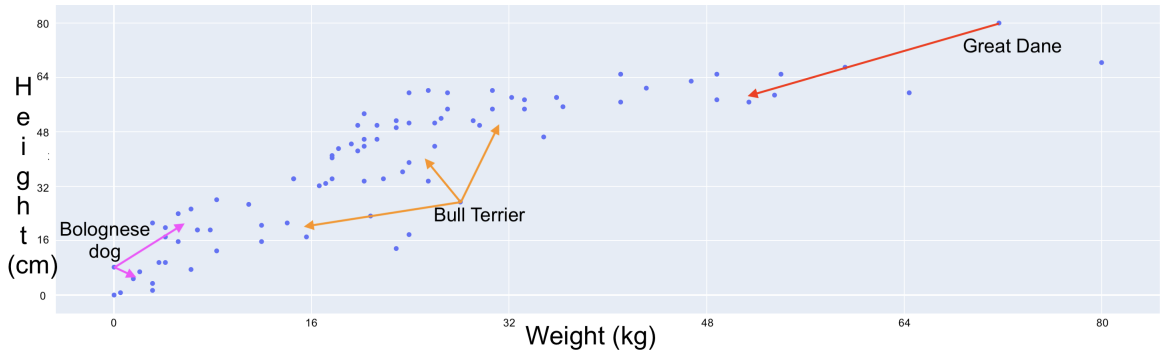


FIGURE 3.2. Explanations for a simple dataset of dog breeds considering their median height and weight and our explanations for why some points are considered anomalous. The Great Dane is anomalous because it is significantly taller and heavier than the other breeds, whereas multiple explanations exist for dogs such as the bull terrier and Bolognese using either height or weight or combinations of the two.

Our work makes the following contributions:

- We explore the understudied (eXplainable Anomaly Detection) XAD problem for both popular classic AD methods and deep AD methods. (See Section 3.3).
- We create a framework with three well-defined steps to create diverse relative explanations as to why the instance is anomalous (See Section 3.4).
- We use our framework for two core tasks: Generating explanations for humans (Table 1) and generating explanations for machines to improve performance (Figures 4, 5).

TABLE 3.1. Comparison of different techniques against our approach. Unlike existing work, our technique is model and task-agnostic and creates explanations that are both contrastive and relative, which we argue is superior for anomaly detection.

Technique	Task Agnostic	Algorithm Agnostic	Contrastive	Relative
Sipple & Youssef 2022 [80]	✓			✓
Clever Hans (Kauffmann et al 2020) [81]	✓	✓		
Diverse Counterfactuals For AD in Time Series. (Sulem et al 2022) [82]		✓	✓	✓
Taylor Decomposition Of One-Class Models (Kauffmann et al 2020) [83]	✓			
PUPAE for Time Series (Der et al 2024) [84]		✓	✓	✓
Shapley Additive Explanations For AD (Antwarg et al 2021) [85]	✓			
Ours	✓	✓	✓	✓

- We empirically demonstrate that our technique creates good explanations for a variety of anomaly detection algorithms, datasets, type of anomaly (global vs contextual, outlier vs novelty), and anomaly score (Figure 3.7).

Our paper is organized as follows: first, we overview existing XAI techniques and why they are not ideal for explaining anomalies and place our work into the context of existing XAD methods. We then discuss desired characteristics for XAD, build a framework to create such explanations, and create an algorithm that uses this framework. We apply our algorithm to the fifteen datasets of the ODDS anomaly detection benchmark dataset collection [79] and demonstrate that the algorithm achieves very little error regardless of the anomaly detection algorithm, anomaly score, or dataset. We further demonstrate our framework's utility through ten simulated user experiments which demonstrate its practicality. Finally, we discuss these results and conclude.

3.2. Related Work & The Need For XAD

Here we overview related work with the intent to highlight the need for a relative explainable anomaly detection framework. We summarize the differences between our work and the existing state of the art in Table 2.1.

Insufficiency of Classic Supervised XAI Methods. Existing XAI for supervised learning, such as LIME [14] and Anchor [75] are popular methods to explain multiclass prediction problems. These methods could be applied to some AD methods that generate an explicit classification function but are unsuitable for the plethora of property-based AD methods. These include many density-based approaches such as LOF [78], Parzen windows, and graph-based methods [86]. Even if XAI-supervised learning methods can be applied to an AD method, the style of explanation is not conducive to explaining anomalies. The classic examples used to justify LIME style explanation are inherently focused on explaining why something belongs to a particular category (i.e. a frog). This makes sense since there are many instances for each category that share an underlying common set of attributes, but when explaining anomalies, these assumptions do not hold. Instead, we typically assume each outlier is unique and there are few of them. This motivates our premise of relative explanations to explain why something is an anomaly by explaining how it differs from normal instances.

Potentially Applicable Supervised XAI Methods. Not all XAI methods explain the output in relation to a heatmap or set of relevant features. Counterfactual explanations [12] directly answer the question of how an instance needs to change to reach some desired output. However, a single counterfactual is ill-suited for the task of explaining anomalies. Consider the bull terrier in the example of Figure 3.2. Here, three explanations can be given for why this breed is anomalous: it is either too heavy for its height, too short for its weight, or some more modest combination of those two factors. No single explanation is sufficient for explaining its anomalous nature, but when considered together the user has a clear understanding of why exactly the bull terrier is anomalous.

Existing XAD Techniques & The Need For Post Hoc Explanation The aforementioned limitations of supervised XAI motivate the study of XAD [87]. Non-relative explanation in anomaly detection tends to use general-purpose XAI techniques for their goals. Chalapathy [88], for instance, cites two attention-map algorithms - algorithms that highlight the most relevant part of the feature space for a particular decision. Other work focused on creating inherently interpretable algorithms [88] [89] [90], rather than providing post hoc explanation. These methods might be appealing in many circumstances, but not all. If other anomaly detection algorithms perform better at a specific task practitioners would be forced to choose between transparency and performance. Further, a group might be interested in explaining the output of an existing AD system where

implementing and testing a new algorithm might be too costly. Other techniques, recognizing the utility of contrastive and/or relative explanations, ground their explanation to a local neighborhood of normal points, however these techniques are either developed for a specific algorithm [83, 85] (most commonly autoencoder-based AD) or for a specific task [82, 84] (time-series data). Our method, by contrast, is the first technique to employ model and task-agnostic while providing contrastive relative explanations. The differences between our technique and existing state-of-the-art XAD are summarized in Table 2.1.

Finally the recent survey on XAD [87] assesses XAD along six dimensions: i) When explanation occurs, ii) What level of granularity is the explanation applied to, iii) Model agnostic or model specific and iv) Feature or sample-based, v) Computation technique used and vi) Applicable to static or streaming data. Our work is an example of a post-hoc, local-level, and agnostic approach to explanation. However, it is quite different from existing work for several reasons. Firstly, unlike existing methods that use the underlying features to explain the anomaly, our work can be applied to entire instances. This means our work can be applied to the results of deep learning methods relatively easily.

3.3. Problem Overview & Definition

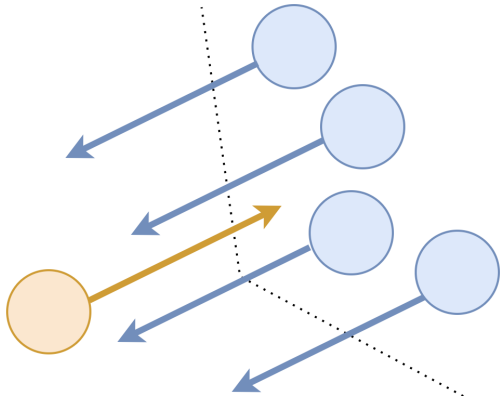


FIGURE 3.3. Visualization of the Definitions 3.3.1 - when the vector is added to the anomaly (orange) it turns normal - and 3.3.2 when the vector is subtracted from the relative normal instance (blue) it turns anomalous. When both of these factors are present, this is a solution to Definition 3.3.3.

A relative explanation attempts to isolate the relevant combination of features that cause an instance to be classified as an anomaly. Specifically, we find a diverse set of changes to the anomaly

such that if those changes are applied to the anomaly it would be considered a normal instance. These changes can be both additive or subtractive modeled as the function g_+ and g_-, respectively. Importantly a positive explanation changes an outlier to a normal point whilst a negative explanation changes a normal point to an outlier.

DEFINITION 3.3.1. ***Positive Relative Explanation Problem.*** *Given an instance x, an anomaly detection function f and an anomaly threshold ϵ, let x be an anomaly i.e. $f(x) > \epsilon$. A positive relative finds a function g_+ which has the following property $f(g_+(x)) \leq \epsilon$.*

DEFINITION 3.3.2. ***Negative Relative Explanation Problem.*** *Given an instance x, an anomaly detection function f and an anomaly threshold ϵ, let x be an inlier i.e. $f(x) \leq \epsilon$. A negative relation explanation finds a function g_- such that for a given anomaly detection function $f()$ we have $f(g_-(x)) \geq \epsilon$ where ϵ is the anomaly threshold.*

In our work, rather than search for $g_+()$ and $g_-()$ separately, we search for an explanation vector (E) that satisfies both the positive and negative explanation qualities. Formally:

DEFINITION 3.3.3. ***Explanatory Vector Problem.*** *Given an anomaly x and a set of close normal points $x'_1 \dots x'_m$, an anomaly detection function f an explanatory vector E for x satisfies the condition $f(x + E) \leq \epsilon$ and $f(x'_i - E) > \epsilon \ \forall i$. That is E is a positive relative explanation for the anomaly x and a negative relative explanation for the nearby normal points x'.*

That is, we find a vector that is a good positive relative explanation when added to the anomaly and a good negative relative explanation when subtracted from nearby normal points. How we construct explanatory vectors is shown in Figure 3.3.

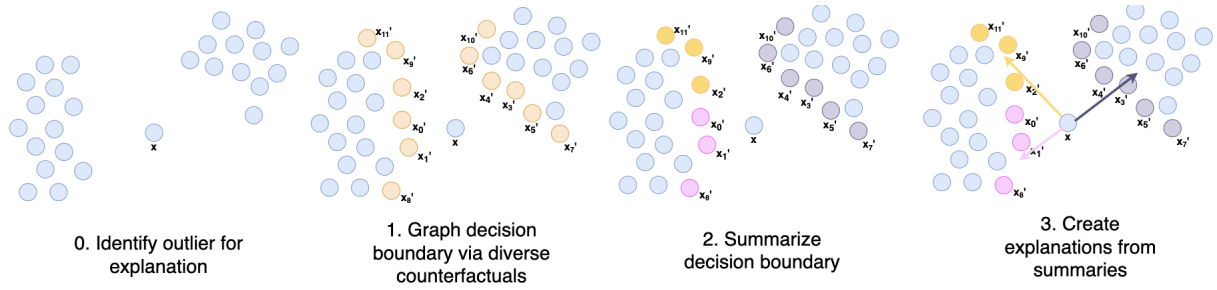


FIGURE 3.4. Pipeline for explanation. First, an anomaly (x, in the picture) is detected by the anomaly detection algorithm (step 0). Then, a diverse set of counterfactual instances (x'0 through x'11) are generated such that they are simultaneously close to the decision boundary, close to the original instance, and distinct from each other (step 2). Finally, patterns are minded from the counterfactuals to extract a set of diverse changes which serve as the final explanation (step 3).

3.4. Our Approach To Finding Explanation Vectors

Our approach has the following steps illustrated in Figure 3.4:

- **Selection:** Identify the anomaly x to explain.
- **Generation:** Create a set of n diverse counterfactuals X' to represent the decision boundary for the AD function f around x. (see Section 4.1)
- **Summarization:** Approximate the decision boundary of f with a subset of representative counterfactuals X^* (see Section 4.2)
- **Generation:** Create multiple explanation vectors $E_1 \dots E_k$ via choosing an appropriate sub-space from the summary X^* (see Subsection 4.2).

The remainder of this section will detail the precise algorithms and optimizations used for each step.

After we are given an anomaly x from any anomaly detection algorithm our approach has three main steps which can be visualized in Figure 3.4. First, we generate diverse counterfactual explanations for the anomaly which are points close to x but not anomalies. This involves a novel counterfactual generation step that comprises a multi-faceted optimization objection shown in equation 3.1. This will generate many counterfactuals, so to avoid overloading the user with possibly redundant explanations we summarize them. To choose a representative set of counterfactuals we formulate and solve a dispersion problem. Finally, we generate a relative explanation from the reduced set of counterfactuals.

3.4.1. Generating Diverse Counterfactuals. Our aim in this step is to take a given anomaly x and produce many diverse counterfactuals. Each counterfactual balances its viability (proximity to the decision boundary, colored blue in equation 3.1), plausibility (distance to the original instance, orange), and diversity (average distance to all other previously generated counterfactuals, magenta). Balance terms are added to the latter two options. Formally, given an anomaly x, an anomaly detection algorithm f, the anomaly threshold score ϵ, and all previously generated counterfactuals CFs, we compute the following:

$$argmin_{x'} \begin{cases} max((f(x') - \epsilon), 0) & \\ +\lambda_1 d(x, x') & \\ +\lambda_2 \frac{\sum_{cf \in CFs} d(x', cf)}{|CFs|} & \text{if } \|CFs\| > 0 \\ \hline max((f(x') - \epsilon), 0) & \\ +\lambda_1 d(x, x') & \text{otherwise} \end{cases} \tag{3.1}$$

This equation is used in an iterative (one counterfactual at a time) manner and is similar in style to the iterative Gonzalez's Farthest-first traversal algorithm [91]. The term in blue is proximity, orange plausibility, and magenta diversity. Diversity is omitted on the first iteration as a single point cannot be diverse.

All counterfactuals are generated via particle swarm optimization [60] (number of particles set to 200) as this is a non-gradient-based optimization technique meaning that it can easily encode the diversity criterion and be used when gradients do not exist, such as in non-neural network-based AD algorithms.

3.4.2. Summarizing and Explaining. The output of the previous step is a set of counterfactuals $x'_1 \dots x'_m$ of which there may be many. In practice, this could be hundreds or even thousands of counterfactual instances. Our next step summarizes them into a set of representative counterfactuals $x^*_1 \dots x^*_n$ where $n << m$ and the final step is to generate an explanation vector between x and each representative counterfactual.

We can formulate the representative counterfactual problem as an instance of the classic computer science dispersion problem [92]. The maximum m-dispersion problem is as follows: given

a collection of points, choose a subset of n points such that the distance between all pairs of chosen points is maximized. Well-known heuristics exist for it that guarantees a constant factor approximation. This effectively takes a large collection of points and summarizes them with n representatives.

The Dispersion Problem.

A given collection of m counterfactuals $X' = \{x'_1 \dots x'_m\}$ chose a subset $X^* = \{x^*_1 \dots x^*_n\}$ $n << m$, $X^* \subset X'$ such that $S = \sum_{i,j} D(x^*_i, x^*_j),\ (x^*_i, x^*_j) \in X^*$ where D is some distance function.

In essence, this summarizes the boundary by grouping counterfactuals such that groups are all different, and the resulting explanation vectors are those that point in the most different directions.

Pattern Mining via Clustering. One can mine frequent patterns in the counterfactuals to find several common trends among the counterfactuals. Because of the continuous state space of most datasets, patterns are mined via KMeans clustering [93], and the explanatory vectors are determined by subtracting the anomaly by the centroid of a cluster.

The Sub-Space Explanation Problem. Given a collection of n representative counterfactuals $X^* = \{x^*_1 \dots x^*_n\}$ solve the optimization problem $argmax_R\ S = \sum_{i,j} D_R(x, x^*_i),\ x^*_i \in X^*$ where R is a subset of features and D_R is some distance function defined on the subspace defined by these features.

In this paper, we examine the validity of all three of these approaches by trying each of them for every anomaly we explain.

3.4.3. Setting Hyperparameters. The optimization of Equation 3.1 uses two balance terms. The first referred to as λ_1 prioritizes similarities between counterfactuals and the original instance and the second, λ_2, prioritizes the diversity of counterfactuals. We propose a principled manner in which one can set these parameters based on common practice in counterfactual research [12]. First, λ_2 is set such that a desired number of diverse explanations exist. Here, two patterns are considered diverse if a certain number of features among the top 10 features changed in both the positive and negative direction are different from one another. Then, tune λ_1 to be the largest value such that at least 95% of counterfactuals generated are considered normal. In this way, the boundaries created by the counterfactual are both accurate and as plausible as possible while

still generating the desired number of diverse explanations. This method provides the freedom for practitioners to gain whatever insights they feel are needed for their particular purpose.

3.5. Experimental Design

3.5.1. A Taxonomy of Anomalies. Before discussing our experiments, we overview the taxonomy of anomaly types [86], and demonstrate that our technique can handle most of these forms of anomalies. Instances are typically considered anomalies if they are either global, exhibiting a particular feature outside of its normal range (ie an eight-foot tall person), or contextual if the values of a particular combination of features together are uncommon [86], such as a five foot tall four-year-old. A global anomaly is typically easier to identify and may not require much in terms of explanation, however, the latter is nontrivial, as high dimensional data will contain many possible combinations of features of various sizes. Here, we leverage innovations in the field of XAI (counterfactual instance generation) to explore how to isolate these features and present them to the user.

As well as being global or contextual, anomalies can also be classified as outliers, anomalies, or novelties [88]. While related, each of these terms describes something different. To illustrate this point, consider a theoretical dataset of dogs with two features per instance: length from tail to nose, and weight. The term "Outlier" is typically used to describe an instance from a population that has distinctive features. Using the dog example from Figure 3.2, an outlier may be a particularly large dog. "Novelty" refers to a new type of instance in a dataset that evolves the distribution over time; for instance, the introduction of a novel dog mix between two breeds which is anomalous at the moment, but will eventually shift the distribution and become the norm. "Anomaly", on the other hand, refers to an instance that comes from a different population from the data sampled and appears different than the normal population, such as a Dachshund if Dachshunds were not well represented in the data, as they are abnormally light for their length. "Anomaly" is also often used as a universal term for these three terms.

Experimental Overview. To test the validity of our approach, we created two experiments to test various aspects of the approach. We must ensure that the desired characteristics outlined in Section 3.3 are satisfied and robust to the choice of anomaly detection algorithm, dataset, and anomaly score. Further, we validate that our explanations are useful to practitioners through a

simulated user experiment in which we demonstrate that the simulated user can identify novelties better after seeing our explanations for them (see Figure 3.8a and Figure 3.8b).

3.5.2. Experiment 1 Design: How Accurate Are The Explanations. Here, we apply our method to four different anomaly detection algorithms, including both classic and deep anomaly detection methods: isolation forest, local outlier factor (LOF), autoencoder-based anomaly detection, and GAN-discriminator-based anomaly detection. We apply each of these techniques to the fifteen publicly available datasets (Lymphography [94], Wisconsin Breast Cancer [95], Cardiography [96], Glass Identification [97], Letter Recognition [98], Mammography [99], Musk [100], Statlog [101], Satlog Shuttle [102], Speech Identification [103], Seismic Bumps [104], and Wine [105] [71]) of the ODDS collection, a standard collection of anomaly detection datasets [79]. Note the ODDs collection gives ground truth for the outliers. For each algorithm and each dataset, we consider the 10 outliers with the highest anomaly score, and the 10 outliers with the lowest anomaly score to demonstrate that our approach can handle both extreme anomalies and borderline anomalies. This implies the creation of 3600 explanations (four algorithms, fifteen data sets, three summarization techniques, and 20 anomalies) for 1200 different anomalies. To measure success, we examine two metrics based on the desired characteristics: anomaly-loss, if or how close the anomaly is to being an inlier after the pattern is applied to it (positive relative explanation), and inlier-loss, if or how close the inlier counterfactuals become to anomalies when the patterns are subtracted from them (negative relative explanation). Anomaly loss is:

$$loss(x_a, f, t) = max(\epsilon - f(x_a), 0) \tag{3.2}$$

And, similarly, inlier loss:

$$loss(c, f, t) = max(f(c) - \epsilon, 0), \forall c \in C \tag{3.3}$$

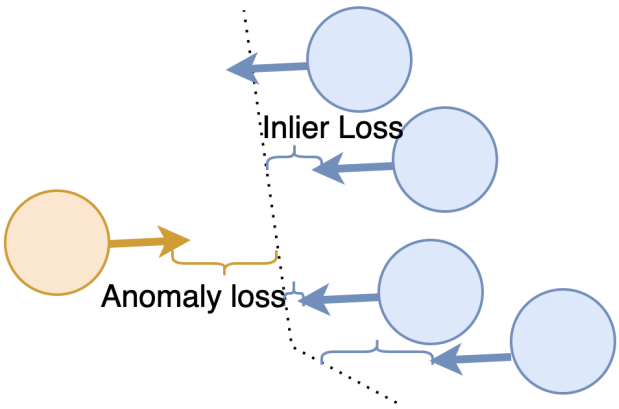


FIGURE 3.5. Visualization of different types of loss. In this example, the explanation vector provided will have some anomaly loss as adding it to the anomaly does not cross the threshold to become normal. The three lower local inliers have some inlier loss because the subtraction of the explanation vector does not turn them all anomalous. The top inlier has zero loss because subtracting the vector from the inlier crosses the anomaly threshold

Where x_a is the anomaly, C is the set of counterfactuals along the decision boundary, f is the anomaly detection function, and ϵ is the threshold for a point to be considered anomalous. This is represented graphically in Figure 3.5. In this way, a method will have zero loss if, when added to any of its patterns, the anomaly is considered normal and, when subtracted from the patterns, inliers are considered outliers, as is claimed in the desired characteristics. These results are shown in Figure 3.7. Further, we present several of these explanations as examples (see Figure 3.9 and Table 3.2) to demonstrate the explainable power of our algorithm.

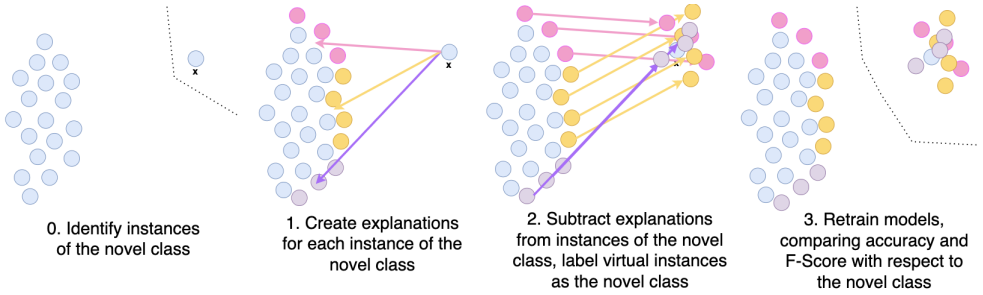


FIGURE 3.6. Pipeline for explaining novelties and measuring explainability. A baseline model is created, anomalies are identified and explainable vectors are generated. Virtual instances are added by subtracting the counterfactuals by their respective vectors, labeled in a self-supervised fashion, and the model is retrained.

3.5.3. Experiment 2 Design: Utility For Novelties. In this simulated user experiment, we demonstrate that our explanations are useful for understanding a novel class. To test this, we

simulate a setting in which we have a series of normal points, but a few new, novel points are then added. We then measure how well our explanations can explain how the novel class differs from the regular class.

To accomplish these tests, we use four datasets (Banknote Authentification [106], Rice Identification [107], Heart Disease [70], and Wifi Localization [108]) with relative class balance (no class has more than 60% representation). We partition the dataset into training and testing sets such that the test set is perfectly balanced, and undersample the "novel" class such that it only comprises 5% of training instances. We bolster the datasets in a self-supervised manner by creating our explanations, adding our explanation vectors to counterfactual instances on the decision boundary, and retraining the models. This process can be visualized in Figure 3.6.

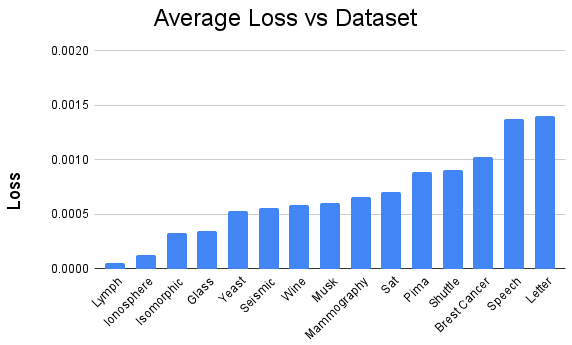
A simulated user in this experiment is a simple decision tree classifier, and to measure our explanation's interpretability, we take a baseline measurement of the simulated user's accuracy and F1 score (with the novel instance being the positive class and all others the negative), a topline of the dataset without undersampling, and record our results training the user on the bolstered dataset. To grant greater validity to our results, we perform the split and the under-sampling five times each, and examine each of the classes as the novel class for a total of 25 trials for each class of each dataset.

Because it is useful to have a topline of a balanced dataset in this experiment, none of the outlier detection datasets listed above would be sufficient for our purposes. Instead, we use the four aforementioned datasets which provide 10 classes. For each trial, we consider one class to be "novel" and the other class(es) to be normal. Only the novel class is undersampled.

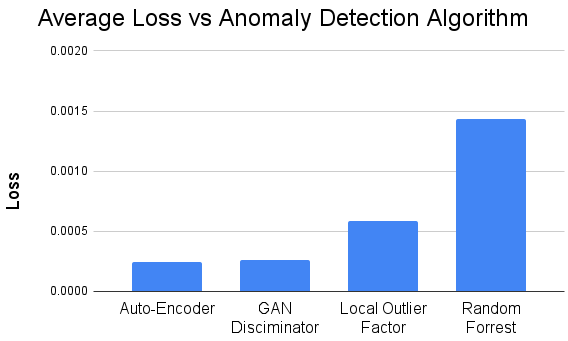
3.5.4. Experiment 3 Design: Visualizing Explanations. In experiment 3, we apply our method to an illustrative computer vision problem using the MNIST [77] handwritten digits dataset. Here, we use a supervised deep anomaly detection technique to graph a manifold for each class and measure anomaly score based on the proximity of the instance to that manifold. This is determined by training a classifier for all ten digits, and the distance from the manifold is calculated by the output logits. If the instance is predicted weakly, that is, two or more logits are highly activated (i.e. it belongs strongly to more than one class), this instance is considered to be between two local manifolds and therefore anomalous.

We use our technique to graph the decision boundary on the manifolds the instance is projected near. Then, we summarize this boundary via the dispersion problem described in Section 3.4.1. The main differences between this experiment and Experiment 1 is that this is used for a computer vision problem and, because we know the anomaly detection algorithm is a deep learner, we optimize via backpropagation to the input space rather than PSO.

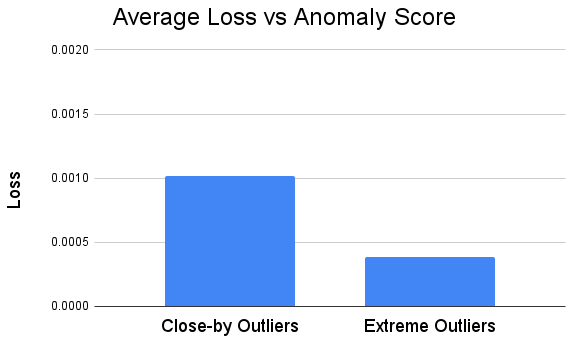
3.6. Experimental Results



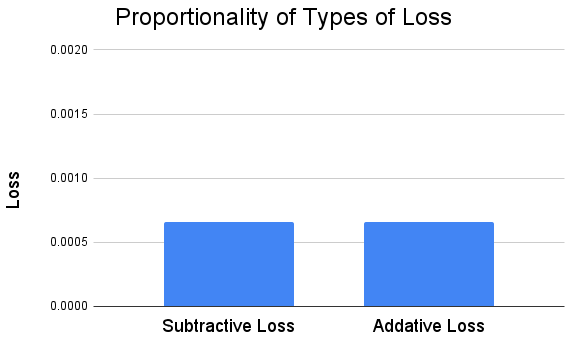
(a) Average (across 20 outliers) loss with respect to each of the fifteen datasets.



(b) Average (across 300 outliers) loss with respect to each of the four anomaly detection algorithms.

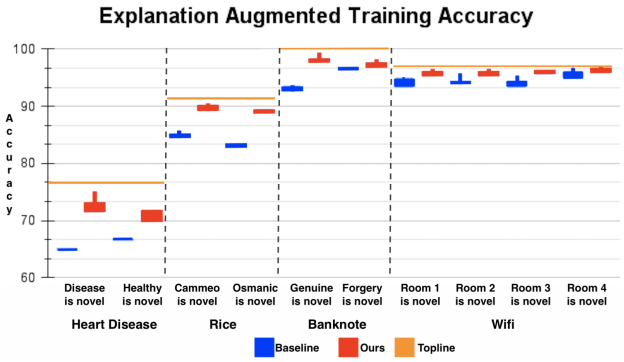


(c) Average (across 1800 explanations) loss with respect to anomaly score - the 10 most borderline outliers and the 10 most extreme outliers.

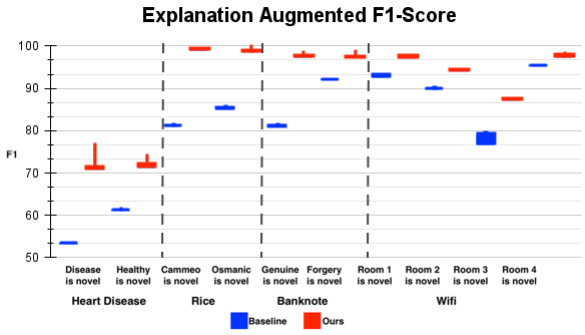


(d) Average (across 1200 explanations) loss, separated into additive and subtractive loss.

FIGURE 3.7. Average loss with respect to different algorithms (a), anomaly detection algorithm (b), and anomaly score (c), along with the proportion of loss that is additive vs subtractive (d). Loss is near zero in any case, and no particular case induces very much error. Note that between the time of experimentation and publication, the Pima Indians Diabetes Dataset has been removed, rendering this particular experiment (but only this experiment) irreproducible.



(a) Accuracy of the simulated user before and after it is provided with our explanation. Accuracy consistently improves towards the topline.



(b) F1 score of the simulated user with respect to the novel class across 10 random samples.

FIGURE 3.8. Experimental results demonstrating the simulated user experiment. Each candlestick represents a trial, which is repeated 10 times. The body of each candlestick represents the lower (25%) and upper (75%) quartiles, and upper and lower shadows represent the minimum and maximum values, respectively.

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Mean	1.52	13.4	2.68	1.44	72.7	0.497	16.2	0.175	0.057
Standard Diviation	0.003	0.817	1.44	0.499	0.774	0.652	8.60	0.497	0.0974
Anomaly 1	0.297	0.372	0.775	0.349	0.505	0.095	0.279	0	0
Explanation 1	0	0	1.26	0.931	0	0	0	0	0
Explanation 2	0	0	0	0	0	0	6.1	0.20	0
Anomaly 2	1.52	14.1	0	2.88	72.6	0.08	8.91	1.06	0
Explanation 1	0	0	0	0	0	0	0	0.29	0
Explanation 2	0	0	0	0	0	0	0	-0.21	0
Anomaly 3	1.52	12.9	1.61	2.17	72.2	0.24	9.7	0.24	0.51
Explanation 1	0	0	0	0	0	0	0	0	-0.18

TABLE 3.2. Feature statistics, examples of anomalies, and their explanation(s) for the glass identification dataset.

In this section, we evaluate our method's interpretability through three experiments detailed in section 3.5. Specifically, we address the following questions:

- Experiment 1: By evaluating 3600 explanations from 300 outliers across 15 datasets 4 anomaly detection datasets, and different anomaly scores, we demonstrate that our method

is sufficient to handle any such setting (see Figure 4). Further, we empirically demonstrate several of our explanations and discuss how they are useful to end-users.

- Experiment 2: In 10 trials over four datasets, we quantify the utility of our algorithm through a simulated user experiment in which the simulated user is tasked with classifying novelties, and is demonstrated to perform better when provided with explanations from our method.
- Experiment 3: We present some of our explanation vectors for our vision experiment on the MNIST dataset.

3.6.1. Experiment 1: How Accurate Are The Explanations. As described earlier, an explanation for an outlier is useful if the patterns returned can turn an inlier into an outlier when subtracted (negative relative explanation) and likewise turn an outlier into an inlier when added (positive relative explanation). Here, we judge our algorithm's capacity to create such explanations by calculating two kinds of error - additive error, that is, the pattern's ability to turn an outlier to an inlier when added, and subtractive error, the ability for a pattern to turn an inlier into an outlier when subtracted (see previous sections and Equations 3.2 and 3.3). To quantify this, we examine our approach using four different anomaly detection algorithms, including two shallow methods (random forest [109] and local outlier factor (LOF) [78]) and two deep methods (autoencoder-based AD [110] and GAN discriminator-based AD [111]) on the fifteen multidimensional datasets of the ODDs dataset collection, a standard outlier detection dataset for benchmarking [79].

As demonstrated in Figure 3.7, the subtractive and additive error is very small irrespective of algorithm, loss, or anomaly score. Even in cases where the loss is highest, such as the letter dataset or under the random forest algorithm, the actual value for loss is very small (less than 0.0015), indicating our method's ability to consistently craft explanations congruent with our desiderata. To put this into perspective, if the anomaly threshold ϵ is set to 0.3, then 0.0015 is within 0.5% of the decision boundary.

Example Explanation Vectors. Here, we examine some of our explanations to determine their interpretability on the glass identification dataset [97]. We examine this dataset specifically because its feature space is easy to understand (RI is the refractive index of the glass, and the other categories indicate the presence of certain chemical elements). Table 1 shows the mean and standard deviation of each feature, along with three anomalies and our method's explanation. The third case is perhaps

the most simple. Here, while most features are roughly in line with regular values, the iron (Fe) levels are extremely high. This instance has the largest value for iron by a wide margin (the second largest having only 0.37), and the explanation brings the instance close to a group of normal high-iron glass samples. The second is perhaps the most interesting. One may notice that the barium levels are significantly larger than other samples, however, the explanation is not a simple subtraction. It includes both a subtraction to barium and an addition. Upon further investigation, there exists a local neighborhood of glass samples (labeled headlamps) that contain high levels of barium. While most instances exist with significantly less barium and a smaller pocket exists with more barium, the anomaly was in a lonely middle ground. Finally, anomaly 1 has two explanations relying on two different sets of features which bring it closer to two different local neighborhoods of points. Our technique not only finds explanations that conform to the general trends of the data but also provides us with insights to understand the full context of why these instances were considered anomalous. This demonstrates that our explanations can explain both contextual anomalies (anomalies 1 and 2) and global anomalies (anomaly 3).

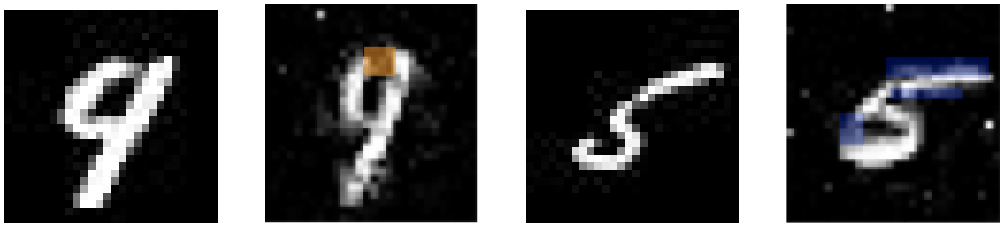
Further results of this experiment can be seen in Figure 3.7, and demonstrate that our explanations can achieve the desired characteristics with near-zero error on any of these datasets irrespective of the base algorithm. Median error across all datasets is less than $2 * 10^{-4}$, and on some algorithms, the median error is zero. Even on the datasets with the highest error, the error is consistently less than 0.0015.

Hyperparameter Settings and Computation Time. We set hyperparameters for this task according to the method presented in Section 3.4.3, and use 100 counterfactuals to graph the boundary. We run these experiments on the anomalies with the top 10 anomaly scores (very anomalous instances) and the 10 anomalies closest to the threshold (borderline anomalous instances) to ensure that our method can handle either case. These anomalies were determined by the four aforementioned algorithms and three different types of explanation (pattern mining, dispersion, and subspace) were generated for each. Therefore, this experiment examines 3600 explanations generated from 1200 anomalies over 15 datasets, and calculates the error for each of them. This was accomplished on a cluster of five Intel Xeon 2.20GHz processors averaging 14 seconds per explanation, for a total compute of 14 CPU hours.

3.6.2. Experiment 2: Explaining Novelties and Data Augmentation. While Experiment 1 demonstrates that our method consistently satisfies our definition of a good explanation regardless of dataset, anomaly detection algorithm, or anomaly score, Experiment 2 quantifies the utility of our explanations and demonstrates their ability to explain novelties.

In this simulated user experiment, we use a decision tree to classify a novel class. While each dataset is initially balanced, we consider one class to be novel and the other(s) normal. The dataset is partitioned into random training and test sets with balanced test sets, and the novel class is undersampled in the training set? until it only comprises 5% of the data. A baseline model is created, the dataset is augmented using our explanation vectors, and the classifier is retrained. Figure 3.8a demonstrates that our method can successfully explain a novel class to a simulated user to a degree very similar to if the simulated user was presented with real instances. Figure 3.8b demonstrates that the novelty is being explained well and completely, whereas 3.8a demonstrates that this is not done at the expense of accuracy in the other classes.

3.6.3. Experiment 3 - Vision. Figure 3.9 demonstrates several explanations for MNIST anomalies, as described in Section 5.4. This provides a visual representation of the explanation vectors explored in the preceding subsections. As one can see from these instances, our technique can find explanations by either removing or adding certain groups of pixels (the explanation vector) such that the image is normal relative to a nearby neighborhood of normal points. Sometimes, as is the case for Figures 3.9f and 3.9h, this requires explanation across multiple neighborhoods for the same instance.



(a) An anomalous nine which looks like a four

(b) Our technique subtracts the unclosed loop of the nine to make it look more like a normal four.

(c) An anomalous five with a nearly looping tail making it look like a six.

(d) Our technique completes the tail and smooths out the head to make the image more like a normal six.

(e) An anomalous nine with a looping tail.

(f) Our technique removes the loop making it look more normal.

(g) The same anomalous nine with a looping tail.

(h) Our technique completes the loop making it look like a normal eight.

FIGURE 3.9. Several anomalies (left) and their visual explanation (right). The most relevant part of the explanation vector is highlighted: orange/blue for subtracted/added pixels.

3.7. Conclusion and Future Work

XAI is an important direction as algorithms replace humans in decision-making as it allows explaining the algorithm's decision. However, existing XAI work for supervised unsupervised learning is not a good fit for AD as these algorithms assume many instances for each category/cluster and each instance in a category/cluster has something in common. However, in AD by definition anomalies are rare and typically each anomaly is unique. Instead, the fundamental question to address in XAD is why is this instance not an anomaly.

To find relative explanations as a driving premise we create an algorithm-invariant approach that finds explanatory vectors for each outlier. An explanatory vector has the property that if it is added to the outlier it converts the outlier to a normal point according to the AD algorithm. Further, if the explanatory vector is subtracted from some normal point it makes them anomalous.

We propose a three-step approach as follows. Step 1 finds diverse counterfactuals which in Step 2 are simplified into a representative set of counterfactuals. Step 3 then generates the relative explanations by comparing the representative counterfactuals.

We have demonstrated the use and flexibility of our approach in two experimental settings. The first shows the effectiveness of our explanatory vectors by measuring the accuracy of the claim that adding and subtracting them from outlier and normal points flip the AD algorithm's prediction on them. The second explores the novel area of machine-to-machine explanation by creating new instances to improve upon performance.

Future work will be divided along two lines: algorithmically and application-wise. Our current framework instantiates each step in a given way, we will explore more efficient and useful ways of achieving each step such as geometric-set cover approaches for the representative counterfactual selection problem and new forms of relative explanation. Application-wise we would like to better explore the area of machine-to-machine explanation for uses of active learning and self-supervised learning.

CHAPTER 4

An Exemplars-Base Approach for Explainable Clustering: Complexity and Efficient Approximation Algorithms

Abstract Explainable AI (XAI) is an important area but remains relatively understudied for clustering. We propose an explainable-by-design clustering approach that not only finds clusters but also exemplars to explain each cluster. The use of exemplars for understanding is supported by the exemplar-based school of concept definition in psychology. We show that finding a small set of exemplars to explain even a single cluster is computationally intractable; hence, the overall problem is challenging. We develop an approximation algorithm that provides provable performance guarantees with respect to clustering quality as well as the number of exemplars used. This basic algorithm explains all the instances in every cluster whilst another approximation algorithm uses a bounded number of exemplars to allow simpler explanations and provably covers a large fraction of all the instances. Experimental results show that our work is useful in domains involving difficult to understand deep embeddings of images and text.

4.1. Introduction

NOTE: A previous version of this paper is published in SIAM SDM24 [38, 112] with co-authors Ian Davidson, Antoine Gourru, Julien Velcin, Peter Walker, and S.S. Ravi.

The area of explainable AI (XAI) tries to make the complex results of an algorithm interpretable by humans. Most work has focused on supervised learning [113], and in particular, *instance-level* explanations such as which parts of an image resulted in a certain prediction [114]. Our work differs from most XAI work in several ways. Firstly, we explore unsupervised learning, and in particular, clustering. Secondly, we seek higher level explanations of the *entire* clustering and not just why an instance was placed in a particular cluster. This is not only an understudied problem, but one where explanation is most needed due to the lack of ground truth annotations (i.e., classes) around which explanations can be built.

Existing work on explainable clustering generates explanations in terms of the underlying features used in the clustering [115, 116, 117]. These methods are not suitable for modern settings that use non-interpretable features such as the two settings which we use to demonstrate our work: clustering of deep embeddings of sentences and images. Our own post-hoc explanation methods also require human interpretable tags [118].

Core Idea. We address the need for explanation by creating an exemplar-based approach to clustering that simultaneously finds clusters of points and exemplars that characterize the clusters. We say that an instance x explains another instance y (or instance x serves as an **exemplar** for instance y) if y falls within ϵ distance of x (i.e., y is within the ball of radius ϵ centered at x). Figure 4.1 provides a simple example of the exemplars for a single cluster.

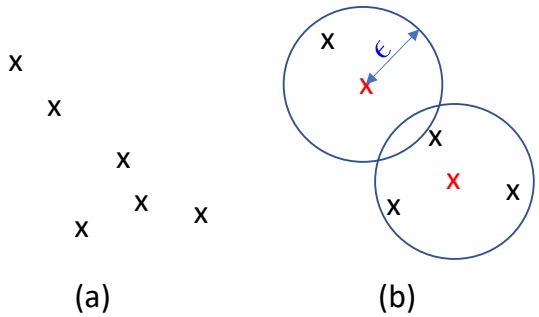


FIGURE 4.1. A simple example of a cluster (a) and the corresponding exemplars (in red) for the cluster (b).

Exemplars are a natural mechanism for the explanation of concepts [37] by enumerating the different variations of the concept. Cognitive science literature (e.g., [119]) indicates that exemplars are ideal for explaining complex concepts/clusters by providing distinct variations of the same concept. However, exemplars cannot be created from existing clustering algorithms. Simply increasing k and using the resultant centroids or finding sub-clusters within clusters [120] does not address this challenge as in many situations there is a natural number of clusters (e.g., Figure 4.3). Further, the variations of the concept need not be dense sub-clusters as shown in Figure 4.3.

Contributions. Our contributions are as follows.

(1) We formulate the novel explainable clustering via exemplars problem and show that even explaining a single cluster is an intractable problem (Theorem 4.4.1).

(2) Our setting is naturally a bi-objective clustering problem with respect to cluster quality and explanation quality but we simplify parameter choice by binding both objectives together with the same parameter ϵ.

(3) We propose a polynomial time clustering algorithm (Algorithm 2) that provides provable performance guarantees with respect to both the maximum cluster diameter and the minimum number of exemplars. More precisely, the maximum cluster diameter is $2(D^* + \epsilon)$, where D^* is the optimal diameter whilst using at most $O(N^* \log n)$ exemplars, where N^* is the minimum number of exemplars needed for the dataset of size n (Theorem 4.4.2). We also present a relaxed version of the algorithm (see Algorithm 3) that upper bounds the number of exemplars by relaxing the requirement to explain every instance.

(4) We experimentally evaluate our methods on several domains involving deep embeddings of images (Faces in the wild), text (a Harry Potter novel) and on MNIST digits. We also begin to explore the novel direction of using exemplars for machine to machine explanation/transfer.

For space reasons, most of the proofs are omitted. They can be found in [112].

4.2. Overview of Our Approach

The input to our method is a collection of instances that we wish to both cluster and explain. Hence, our method is an example of an explainable-by-design clustering algorithm, unlike our previous work that attempts to find an explanation for a given clustering [118]. Further, unlike prior work on conceptual clustering, we do not use the features used to cluster in developing an explanation; for instance, the work discussed in [115] simultaneously builds a clustering and a decision tree using the same features. Here, we instead find a clustering and a suitable subset of the instances (which we call exemplars) within each cluster to explain it. We say an exemplar explains a set of instances that are within ϵ distance of it. In practice, exemplars are significantly different from cluster centroids; see Figure 4.2 for an example.

Trading Off Explanation Complexity Against Clustering Quality. We design clustering algorithms that ensure that the maximum diameter of the clustering found is within a small constant factor of the optimal diameter and ϵ (the radius of an exemplar's coverage). Hence, the parameter ϵ provides a natural way to trade off explanation complexity against cluster compactness. If we make ϵ small, we naturally will require more exemplars but will find more compact clusters. Conversely,

if we make ϵ large, we will create simpler explanations but at the cost of a larger cluster diameter. We present efficient approximation algorithms that provide provable performance guarantees with respect to both the maximum diameter and the number of exemplars used.

Exemplars for Explanation and Their Benefits. Our work can be considered as a quantification of the exemplar-based school of concepts [37] as we are discovering concepts (the clusters) and the exemplars that typify/explain them. This contrasts with feature based explanations (e.g., using the attributes/properties of the face) as described earlier. In this paper, we argue that using exemplars has pragmatic and pedagogical benefits. As ML/DM progresses to more complex representations of complex objects, using features as the basis for explanation is no longer always valid, even though there is excellent work in this area [115]. In settings where features are not interpretable (e.g., deep embeddings of image data), one pragmatic explanation mechanism is exemplars. The pedagogical benefit stems from cognitive psychology's experimentally-verified rich literature on how humans understand and comprehend the world; this literature comes under a topic known as Concept Theory [37, 121].

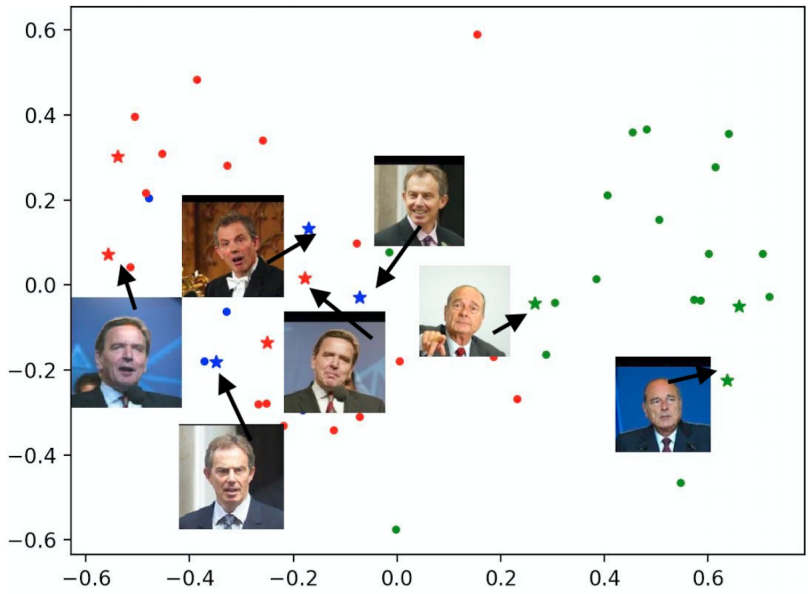


FIGURE 4.2. An illustrative example of generating clusters (color) and selecting exemplars (stars). The exemplars form a prototypical explanation of a cluster in that they cover all instances in the cluster. Note the exemplars need not be (and rarely are) close to the centroids.

Difficulty of the Problem. Our computational problem inherently has two intertwined tasks: (i) finding compact clusters and (ii) finding a minimal set of exemplars to represent each cluster. This is a challenging problem as the first problem is known to be NP-hard [122] and additionally we show that even for a single cluster, finding a minimal set of exemplars to represent the cluster is NP-hard (Theorem 4.4.1). Solving these tasks separately could yield sub-optimal results; instead, we bind them together using a single parameter ϵ (the exemplar coverage distance) to simultaneously perform clustering and exemplar selection. Our algorithms provide provable performance guarantees.

4.3. Definitions

4.3.1. Basic Definitions. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n instances. We assume that for each pair of instances x_i and x_j, we have a (symmetric) distance $d(x_i, x_j)$. The distance function d is assumed to be a *metric*; it may be the distance in some embedding space. We are also given a value $\epsilon > 0$ which is set by a domain expert and naturally trades off explanation complexity against cluster compactness as discussed earlier.

Notion of Explanation. Given two instances x_i and x_j, where $d(x_i, x_j) \leq \epsilon$, we say that x_i covers x_j and that x_i is an exemplar for x_j and vice versa (since d is symmetric). For convenience, we will also say that x_i is an *ϵ-neighbor* of x_j (and vice versa). We now generalize this definition to clusters. Given a subset $Y \subseteq X$ of instances and another subset $\mathcal{E} \subseteq X$ of exemplars, we say that $\mathcal{E}$ covers Y if for every instance $x_i \in Y$, there is an instance $x_j \in \mathcal{E}$ such that x_j covers x_i (i.e., x_j is an exemplar for x_i). When a subset $\mathcal{E} \subseteq X$ of exemplars covers a set $Y \subseteq X$, we say that $\mathcal{E}$ is an exemplar set for Y and that $Y \cup \mathcal{E}$ forms a cluster and its explanation.

For any instance x_i, let $S_i \subseteq X$ consist of all the instances that are ϵ neighbors of x_i ($d(x_i, x_j) \leq \epsilon$). We refer to S_i as the *ϵ-neighborhood* of x_i.

Clustering to Minimize the Maximum Diameter. For clustering a set X of instances, a common objective is to minimize the maximum diameter [122]. For the reader's convenience, we provide the associated definitions. The diameter of any cluster is the maximum distance between any pair of instances in that cluster. The diameter of a clustering is the largest cluster diameter. It is known that finding a clustering with $k \geq 3$ clusters that minimizes the maximum diameter is NP-hard [123]. When the distance function is a metric, a well-known approximation algorithm

due to Gonzalez [122] provides a clustering whose maximum diameter is at most twice the optimal diameter. This paper by Gonzalez also shows that unless **P** = **NP**, no efficient algorithm can provide a better performance guarantee.

Minimum Set Cover (MSC) Problem. In this problem [123], the input consists of a base set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $Y = \{Y_1, Y_2, \ldots, Y_m\}$, where each Y_j is a subset of U $(1 \leq j \leq m)$ and an integer bound $\beta \leq m$. The goal is to choose a subcollection Y' of Y with $|Y'| \leq \beta$ such that the union of the sets in Y' is equal to U (i.e., the union covers all the elements in U). This problem is NP-Complete and a natural greedy approximation algorithm (which picks a new set in each iteration such that the set covers as many new elements as possible) is known to give a performance guarantee of $O(\log n)$ for the problem [124]. It is also known that under well accepted hypotheses in complexity theory, there can be no better polynomial time approximation algorithm [125]. One of our results (Section 4.4.3) uses this greedy approximation algorithm.

Budgeted Maximum Coverage Problem. We also use a known approximation algorithm for the Budgeted Maximum Coverage (BMC) problem, which is closely related to the Minimum Set Cover (MSC) problem [123]. The input to the BMC problem is a base set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $Y = \{Y_1, Y_2, \ldots, Y_m\}$, where each Y_j is a subset of U $(1 \leq j \leq m)$ and a budget $\beta \leq m$. The goal is to choose a subcollection Y' of Y with $|Y'| = \beta$ such that the union of the sets in Y' covers the maximum number of elements of U. This problem is also NP-hard and a natural greedy approximation algorithm (which picks a new set in each iteration such that the set covers as many new elements as possible) has been shown to give a performance guarantee of $(1 - 1/\mathrm{e})$ for the problem [126], with e being the base of the natural logarithm. We use this result in Section 4.4.3. It is also known that under well accepted hypotheses in complexity theory, the performance guarantee of $(1 - 1/\mathrm{e})$ cannot be improved [125].

We also use some standard graph theoretic notions (such as dominating sets and unit disk graphs). These definitions can be found in [112].

4.3.2. Main Problem Formulations. We now provide rigorous formulations of the problems considered in this paper. We begin with the problem of finding a small set of exemplars for a given set of instances.

(a) **Minimum Set of Exemplars for a Cluster** (MSEC)

Given: A cluster $X = \{x_1, x_2, \ldots, x_n\}$ of n instances, a value $\epsilon > 0$, an integer $\beta \leq |X|$.

Question: Is there a subset $\mathcal{E} \subseteq X$, with $|\mathcal{E}| \leq \beta$, such that $\mathcal{E}$ is an exemplar set for X?

We note that the MSEC problem requires an exemplar set for *all* the instances in the set X. We now develop formulations where the set X must be partitioned into clusters and exemplar sets must be found for each cluster. We first provide a formulation where each instance must have an exemplar.

(b) **Simultaneous Construction of Clusters and Exemplars** (SCCE)

Given: A set $X = \{x_1, x_2, \ldots, x_n\}$ of n instances to be clustered, integer k, where $2 \leq k \leq n$ (the number of clusters), and a value $\epsilon > 0$.

Requirement: Find a partition of X into k clusters C_1, C_2, ..., C_k and an exemplar set $mathcalE_j$ for each cluster C_j, $1 \leq j \leq k$, such that all the following conditions hold:

- Compactness of Clustering and Explanation: (i) the maximum diameter of the clusters is as small as possible, (ii) $\sum_{j=1}^{k} |mathcalE_j|$ (i.e., the total number of exemplars used) is as small as possible.
- Distinctness of Explanations: (iii) $mathcalE_a \cap mathcalE_b = \emptyset$ for all $1 \leq a, b \leq k$ and $a \neq b$ (i.e., the exemplar sets are pairwise disjoint), and
- Completeness of Explanations: (iv) for each instance $x \in X$, there is an exemplar y such that x and y are in the same cluster.

We present an approximation algorithm for SCCE in Section 4.4. However, this solution may use a large number of exemplars due to the completeness requirement. This can make it difficult for a user to interpret the explanation. To address this, we next explore a relaxed version of the problem where not all instances are explained. (Our approximation algorithm for this problem allows us to get a lower bound on the number of instances that are explained.)

(c) **Simultaneous Construction of Clusters and β-Bounded Exemplars** (SCCRB)

Given: A set $X = \{x_1, x_2, \ldots, x_n\}$ of n instances to be clustered, integer k, where $2 \leq k \leq n$ (the number of clusters), a value $\epsilon > 0$ and integer β (upper bound the total number of exemplars for all clusters).

Requirement: Find a partition X into at most k clusters $C_1, C_2, \ldots, C_k$ and the corresponding exemplar sets $mathcalE_1$, $mathcalE_2$, ..., $mathcalE_k$ as in the SCCE problem above with the

requirements for compactness of clusters (Condition (i)), distinctness of explanation (Condition (iii)) but now:

- Upper Bound on the Number of Exemplars: (ii) $\sum_{j=1}^{k} |mathcalE_j| \leq \beta$, and
- Relaxing the Condition that Every Instance be Explained: (v) The number of instances which have an exemplar in the same cluster is as large as possible.

We present an approximation algorithm for SCCRB in Section 4.4. Note that compared to SCCE, not every instance will be explained (i.e., covered by an exemplar). The algorithm can identify unexplained instances; this could be useful since such instances may represent anomalies.

4.4. Algorithmic Results

4.4.1. Finding a Minimum Set of Exemplars For A Single Cluster. We begin with a complexity result for the Minimum Set of Exemplars (MSEC) problem for a single cluster. As can be seen from our proof in [112], this complexity result holds even when the given set of instances X consists of points in 2D-Euclidean space.

THEOREM 4.4.1. *The MSE problem is NP-hard even when the set of instances X consists of points in 2D-Euclidean space and the distance between any two points is their Euclidean distance.*

4.4.2. An Approximation Algorithm for SCCE. The SCCE problem requires us to find a clustering where the diameter of each cluster and the number of exemplars are as small as possible. Recall that each of these problems is computationally intractable. We present an algorithm that provides a provable performance guarantee for each of these measures.

Overview of the algorithm. First, the algorithm takes the set X and produces pairwise disjoint blocks B_1, B_2, ..., B_k to minimize the maximum diameter [122]. It then uses a greedy approximation algorithm for the Minimum Set Cover (MSC) problem [124] to find a near-minimal set of exemplars A for the set X. For each cluster C_j, the exemplar set $mathcalE_j$ is given by $mathcalE_j = B_j \cap A$, $1 \leq j \leq k$. Finally, each cluster C_j consists of the exemplar set $mathcalE_j$ and all the non-exemplars covered by $mathcalE_j$. This ensures that the exemplars are pairwise disjoint and that each non-exemplar is covered by an exemplar in the same cluster. Note that we only move non-exemplars from their original blocks (i.e., B_1, B_2, ..., B_k) to new clusters (i.e., C_1, C_2, ..., C_k). This is crucial to ensure the performance guarantee on the maximum diameter. An

Algorithm 2 Approximation Alg. for SCCE

procedure ApproxSCCE(X, k, ϵ)
Input: A set of instances X, the number of clusters k, and the exemplar distance bound ϵ.
Output: A clustering of X into k clusters and a set of exemplars for each cluster to satisfy the requirements of the SCCE problem.
Block Creation. Use Gonzalez's approximation algorithm [122] to obtain k (disjoint) blocks $B_1, B_2, \ldots, B_k$ of X.
Exemplar Neighborhood Set Construction. For each $x_i \in X$, find S_i, the set of all instances $x_j \in X$ such that $d(x_i, x_j) \leq \epsilon$. (Thus, x_i can serve as the exemplar for each instance in S_i.)
Exemplar Selection. Construct the Minimum Set Cover (MSC) problem consisting of the base set X and the set collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$. Use a greedy approximation algorithm for MSC [124] to construct a near-optimal set cover given by the subcollection $\mathcal{S}_1 \subseteq \mathcal{S}$. Obtain the exemplar set A as follows: for each $S_i \in \mathcal{S}_1$, add x_i to A.
Cluster Creation. Create k empty clusters $C_1, C_2, \ldots, C_k$.
Exemplar Assignment. For each cluster C_j, the set $\mathcal{E}_j$ of exemplars is given by $\mathcal{E}_j = B_j \cap A$. Add $\mathcal{E}_j$ to C_j.
Non-Exemplar Assignment. Consider each cluster C_j. For each exemplar $x_i \in C_j$, add each instance in $S_i - A$ (i.e., each non-exemplar in S_i) to C_j.
Output the set of clusters C_1, C_2, ..., C_k and the corresponding exemplars $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_k$.

outline of our approximation procedure is shown as Algorithm 2. Note that if an instance x is covered by multiple exemplars, it can be assigned to any cluster that has an exemplar for x. The following theorem shows the performance guarantee provides by Algorithm 2.

Theorem 4.4.2. *The solution produced by Algorithm 2 satisfies the following properties: (i) The diameter of each cluster is at most $2(D^* + \epsilon)$, where D^* is the optimal diameter for a k-clustering of X and ϵ is the exemplar distance. (ii) Every instance in X has an exemplar within the same cluster. (iii) The sets of exemplars for the k clusters are pairwise disjoint. (iv) The total number of exemplars generated by the algorithm is at most $O(N^* \log n)$, where N^* is the minimum number of exemplars needed to cover all the instances in X.*

Proof: See [112].

Remark: Since Step 3 in Algorithm 2 uses an approximation algorithm for MSC, the performance guarantee with respect to the number of exemplars is $O(\log n)$, where $n = |X|$. Theoretically, one can get a better performance guarantee (namely, $(1 + \delta)$ for any fixed $\delta > 0$) with respect to the number of exemplars while ensuring that the maximum diameter is at most $2(D^* + \epsilon)$. This is done by transforming the Exemplar Selection steps (i.e., Steps 2 and 3 of the algorithm) into that of finding a near-optimal dominating set for unit disk graphs in an Euclidean space whose dimension ℓ is the same as that of the points in X. However, this approximation algorithm (whose running

time includes the factor $O(n^{(1/\delta)^2})$ [127]) is impractical even for data sets of moderate size. For example, if $\delta = 0.5$, the running time of the algorithm includes the factor $O(n^4)$. For this reason, we decided to use the MSC-based approximation algorithm in our experiments.

Running time of Algorithm 2: The overall running time of Algorithm 2 can be shown to be $O(n^2)$. Please see [112] for details.

4.4.3. An Approximation Algorithm for SCCRB. When ϵ is small, our approximation algorithm for SCCE generates a solution with a small cluster diameter; however, it may yield a large number of exemplars leading to an overly complicated explanation. The goal of SCCRB is also to find a clustering with a small maximum diameter but we relax the requirement to have exemplars for **all** the instances. Instead, we are given an upper bound on the total number of exemplars for all clusters, and we want to maximize the number of instances with exemplars subject to the bound on the number of exemplars.

We now present an approximation algorithm that provides a provable performance guarantee for the diameter as well as the number of instances covered by exemplars in each cluster. This algorithm is similar to the one for the SCCE problem (Algorithm 2) except that it uses a known approximation algorithm for the Budgeted Maximum Coverage (BMC) problem [126] in Step 3 instead of the approximation algorithm for the MSC problem. The steps of this approximation algorithm are shown as Algorithm 3. The following theorem (proved in [112]) establishes the performance guarantee provided by the Algorithm 3.

THEOREM 4.4.3. *The solution produced by Algorithm 3 satisfies the following properties: (i) The diameter of each cluster is at most* $2(D^* + \epsilon)$*, where* D^* *is the optimal diameter for a* k*-clustering of* X *and* ϵ *is the exemplar distance. (ii) The sets of exemplars for the* k *clusters are pairwise disjoint. (iii) The total number of instances with exemplars is at least* $(1 - 1/\mathrm{e})Q^*$*, where* e *is the base of the natural logarithm and* Q^* *is the maximum number of instances in* X *that can have exemplars under the constraint that the total number of exemplars is at most* β*.*

Running time of Algorithm 3: The running time of Algorithm 3 can be shown to be $O(n^2)$. The details appear in [112].

Algorithm 3 Approximation Alg. for SCCRB

procedure APPROXSCCRE(X, k, ϵ, β)
Input: A set of instances X, the number of clusters k, the exemplar distance bound ϵ, and an upper bound β on the total number of exemplars for all clusters.
Output: A clustering of X into k clusters and a set of exemplars for each cluster to satisfy the requirements of the SCCRB problem.
Block Creation. Same as Algorithm 1.
Exemplar Neighborhood Set Construction. Same as Algorithm 1.
Exemplar Selection. Construct the Budgeted Maximum Coverage (BMC) problem consisting of the base set X, the set collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, and the budget β. Use the greedy approximation algorithm for BMC [126] to construct a subcollection $\mathcal{S}_1 \subseteq \mathcal{S}$. Obtain the exemplar set A as follows: for each $S_i \in \mathcal{S}_1$, add x_i to A.
Cluster Creation. Same as Algorithm 1.
Exemplar Assignment. Same as Algorithm 1.
Non-Exemplar Assignment. Consider each cluster C_j. For each exemplar $x_i \in C_j$, add each instance in $S_i - A$ (i.e., each non-exemplar in S_i) to C_j. The set of instances X' which don't have exemplars is given by $X' = X - \bigcup_{S_i \in \mathcal{S}_1} S_i$.
Output the set of clusters C_1, C_2, ..., C_k and the corresponding exemplars $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_k$.

4.5. Experiments

Code and data to reproduce and document the experiments are available[1] with the exception of the Harry Potter novel data which is not in the public domain but is freely available. We have tried to quantitatively and qualitatively evaluate our approach's usefulness for explanation to a human. We explore several directions including generating summaries of a novel which we compare against human written summaries. Similarly, we explore quantitative measures on human faces in the wild data, and for completeness, a qualitative analysis of a standard digit data set. Finally, in an emerging direction of using explanation for machines (not humans), we explore using exemplars for SVM transfer learning.

A discussion on the time used by our algorithms on some data sets is provided in [112].

4.5.1. Qualitative Experiments on Digits Data. Here, we take the standard MNIST data set consisting of 10,000 written digits. We embed them using tSNE [128] and use our algorithm to cluster them and generate exemplars. Our hope is that the exemplars will be a varied representation of the different ways of writing each digit. The clusters found by our methods and approximate centroids (not exemplars) are shown in Figure 4.3. (A larger version of the figure appears in [112].) For each cluster, we present the exemplars found in Table 4.1. Of course, the clustering does not have 100% accuracy but we see that for well separated clusters (0, 5, 6, 7, 8 and 9), the exemplars

[1]URL: www.cs.ucdavis.edu/~davidson/SCCE-DMKD-main.zip. All code and public data are located at the site.

do indeed capture a variety of ways that the digits are written. Quite surprisingly, many are fundamentally different from the centroid. Take for example the digit 7. The centroid has the top line pointing downwards but the exemplars show examples where the top line is up and the vertical line is crossed.

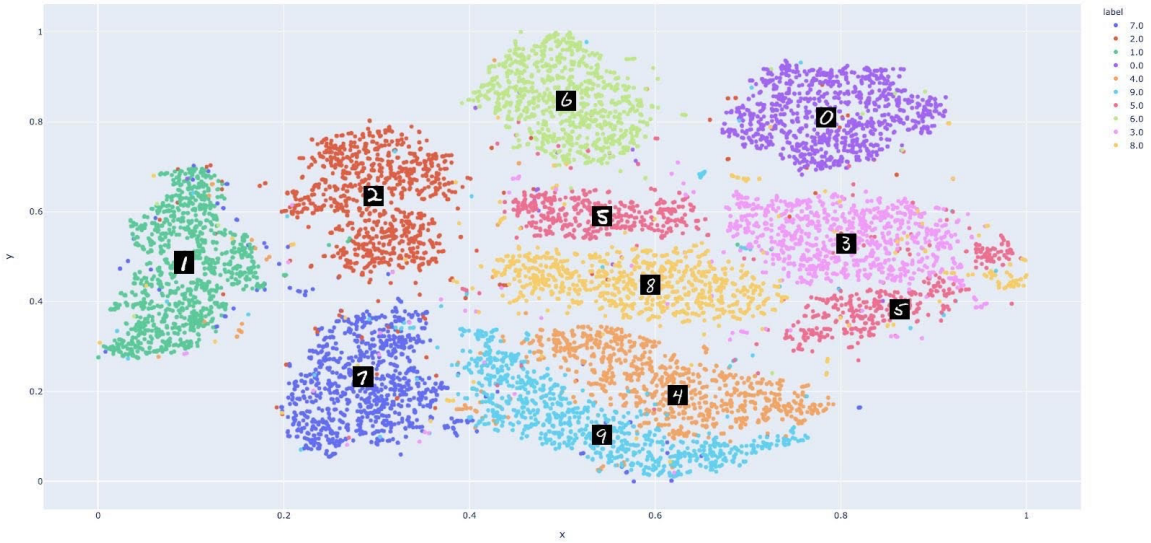


FIGURE 4.3. Clusters and centroids (not exemplars) found by our method when applied to the MNIST dataset. A larger version of the figure appears in [112].

4.5.2. Quantitative Experiments on Textual Data. In this section, we evaluate the ability of exemplars to simplify a corpus by summarizing content. In particular, we take the sentences in the first Harry Potter (HP) book, embed them using deep learning, apply Algorithms 2 and 3 and concatenate the resultant exemplars to form a summary. This is compared with a ranking based approach [129, 130] which can be viewed as choosing exemplars from a list based on importance. These ranking methods are known to produce superior results for HP books [131] compared to recent methods. We measure results by comparing against four human written summaries.[2] Results (Table 4.2) show that our method performs better than these ranking methods by 12.8% and the baseline of random selection of sentences by over 20%. Most importantly, our method's summary score is almost comparable (on average) to the similarity between the human summaries themselves (Table 4.2).

We measure performance using the ROUGE score [132] which is a standard method of evaluating the similarities between computer generated summaries and human written summaries. We

[2] www.britannica.com, en.wikipedia.org, harrypotter.fandom.com, content.time.com

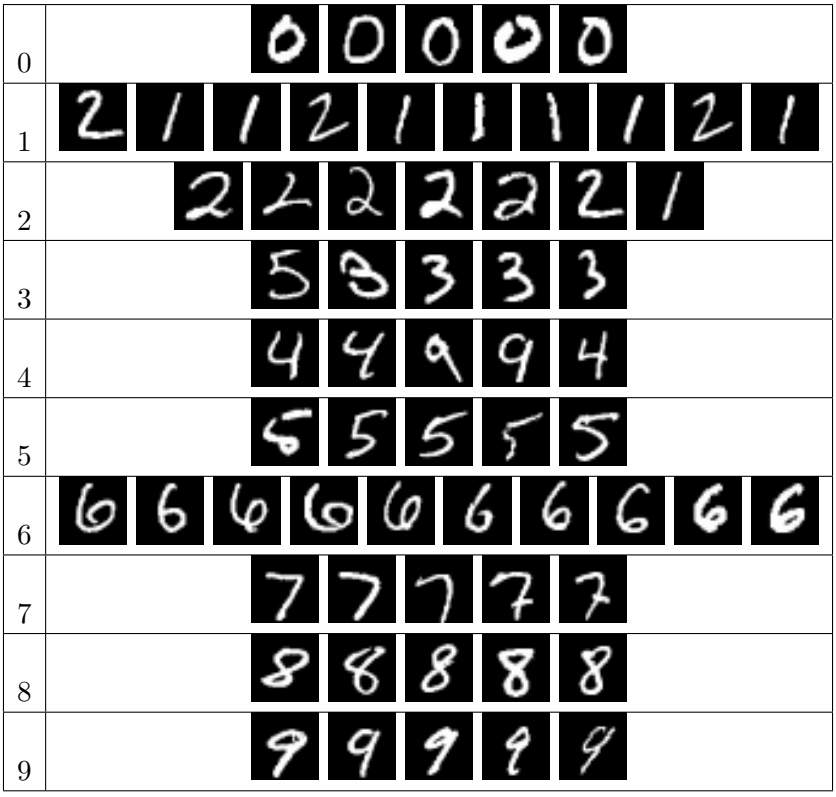


TABLE 4.1. The clusters and exemplars found by our method on the MNIST data set. Note that the exemplars provide a variety of ways that the digits are written and most importantly are quite different from the centroids shown in Figure 4.3.

represent each sentence in the first HP book using the state-of-the-art language model BERT [133]. Hence, the exemplars generated by our method will be sentences in the book. Specifically, we used a fine-tuned pre-trained BERT-base model (https://huggingface.co/).

We compared our two methods against two approaches. The first one is a random subset of sentences used as a control. We repeat this random selection process 20 times. The second baseline is the widely used ranking approach for extracting summaries [129, 130]. These methods require a graph which we construct from pairwise cosine similarities using the sentence embedding obtained with our fine-tuned BERT model. This is a time tested method with thousands of citations and in 2020 still produces state of the art results for the HP literature [131]. For all methods except for SCCE, we fix the number of sentences extracted to be equal to the number of sentences in the ground truth summaries. We use 6 clusters chosen after hyper-parameter tuning to find the stablest clusters. See [112] for an example summary.

	Methods					Relative Performance	
	SCCE (Ours)	SCCRB (Ours)	Ranking [130]	Random	Other Summaries	To Ranking	To Other Summaries
Sum-1	31.65	**33.81**	28.86	23.69	38.09	+17%	-11%
Sum-2	29.58	**31.26**	28.73	25.89	25.82	+9%	+21.1%
Sum-3	27.08	**28.78**	26.58	22.68	31.32	+8.3%	-8.1%
Sum-4	33.33	**34.11**	28.31	24.29	36.08	+17%	-5.5%

TABLE 4.2. The ROUGE-1 F1-scores (the larger the better) measuring the similarity of our two methods, one state of the art (Ranking), one baseline (Random) to four human written summaries (one per row) of the first Harry Potter novel. For each summary, we also report the the average similarity to the remaining three summaries (Human Summaries). Each computational method (except SCCE) generates the same number of sentences as the summary against which it is compared.

Clustering Artifact Used	**Accuracy**
Exemplars	48.33
Cluster centers	44.00
All Points	42.00
Random Points in cluster	44.66

TABLE 4.3. Measuring the effectiveness of exemplars to explain/predict a person from images. Competing methods use the same clustering we find but instead use k-Nearest-Neighbor for prediction with different aspects/artifacts of the cluster. The value ϵ is tuned and set to 0.6 to maximize the stability of clusters.

4.5.3. Quantitative Experiments on Facial Data. One way to determine whether an explanation is useful is to check if it helps a human to understand the underlying concepts which are the clusters. A typical test of exemplar theory given to humans [37] is the task of identifying several people they have never seen before using only a small set of exemplars of the people. We make the task challenging by choosing three similar men (Gerhard Schröder, Jacques Chirac and Tony Blair) and use just 40 images of each person with rest used for testing. See [112] for these results.

4.5.4. Exemplars for Machine To Machine Explanation. Our exemplar and clustering discovery method can also help to explain a problem to a machine. Essentially, our method identifies clusters of points and important examples of each cluster. Here we use those important examples to do instance transfer learning for support vector machines (SVMs). Transfer learning uses a source task to help a target task. We use the well known pendigits dataset [134] to transfer the task of predicting between two digits to help another task of predicting between two very similar digits.

(Source) Target	No Transfer	Transfer Support Vectors	Transfer Exemplars
(1 vs 9) 1 vs 7	0.79	0.83	0.91
(2 vs 8) 2 vs 3	0.78	0.81	0.89
(3 vs 8) 3 vs 9	0.81	0.84	0.88
(1 vs 7) 1 vs 9	0.82	0.83	0.90
(2 vs 3) 2 vs 8	0.80	0.81	0.91
(3 vs 9) 3 vs 8	0.73	0.70	0.89
(1 vs 9) 1 vs 7	0.63	0.72	0.80
(2 vs 8) 2 vs 3	0.64	0.73	0.81
(3 vs 8) 3 vs 9	0.66	0.72	0.81
(1 vs 7) 1 vs 9	0.62	0.71	0.79
(2 vs 3) 2 vs 8	0.61	0.69	0.83
(3 vs 9) 3 vs 8	0.59	0.63	0.77

TABLE 4.4. Accuracy for Transfer Learning. 350 training instances of each digit were randomly chosen for both source and target problems. The 3rd column shows transferring the support vectors and the 4th column shows transferring the exemplars from our work. Results are averaged over 100 random trials. Results above (below) the double lines use all 8 pairs (first 4 pairs) of coordinates. Using just half the features produces nearly twice as many support vectors.

For example, we can learn the source task of `1 vs 9` and transfer it to help the `1 vs 7` task as shown in Table 4.4.

Recall that with an SVM, the vector $\mathbf{w}$ implicitly defines the hyperplane and a constraint to separate the two classes as shown below in Equation (4.1). A common method of performing SVM transfer learning is to add another constraint to the problem that requires the hyperplane to also separate the classes in the source problem. Note the last constraint of Equation (4.1) has (x_i^s, y_i^s) which are the support vectors from the previously solved SVM for the source problem. In our experiments, we instead transfer the exemplars. We use the bounded version of our formulation to transfer over the same number of instances as support vectors in the source problem. Results in Table 4.4 show promise and a future direction of exemplars augmenting existing ML tasks.

$$\text{argmin}_{\mathbf{w},\, w_0} \frac{1}{2}||\mathbf{w}||^2 \tag{4.1}$$

$$\begin{aligned} \text{s.t.} \quad y_i(\mathbf{w}^T.\mathbf{x}_i + w_0) &\geq +1 \ \forall i \text{ and} \\ y_j^s(\mathbf{w}^T.\mathbf{x_j^s} + w_0) &\geq +1 \ \forall j \end{aligned}$$

4.6. Related Work

Explanation and Clustering. The machine learning community has studied explaining clusters from two perspectives. The one-view approach of conceptual clustering [115, 116, 117] proposes a task that is similar to our own (i.e., finding a clustering and its description), but requires that the features used to perform clustering are human interpretable. More recent work [118, 135] has explored post-hoc explaining a *given* clustering using a set of auxiliary tags; it does not find a clustering itself.

Comparison to DBSCAN and Other Density Based Clustering Methods. Superficially, our method may seem to be similar to DBSCAN [136] and other similar algorithms as it uses notions such as ϵ-neighbors. However, there are several fundamental differences. Firstly, our method is guaranteed to use the specified number or near-minimum number of exemplars, where as DBSCAN, while being a very useful method, does not provide such guarantees. Similarly, our method has an explicit clustering objective (i.e., to minimize the maximum cluster diameter) where as DBSCAN does not. Finally, DBSCAN is a not designed so that the core points can be considered explanations of the clusters. As a consequence, it is not meaningful to compare our method with DBSCAN.

The work on multiple centroid methods (e.g., [120, 137]) may appear to be similar; however, they are not explanation focused methods. For more details, see [112].

4.7. Conclusions

XAI for clustering is an under-studied problem compared to supervised learning. Here we explore a style of explainable-by-design algorithm that simultaneously finds clusters and exemplars to describe those clusters. The idea of using exemplars has several benefits. Firstly, it has pedagogic benefits in that humans are known to naturally understand concepts in terms of exemplars [37]. How humans naturally cluster and then organize these exemplars into hierarchical structures will motivate future work. Secondly, the use of exemplars is perhaps the only way to explain data when it is clustered in high dimensional uninterpretable spaces such as deep embeddings. We show that finding a small set of exemplars for just one cluster is NP-hard and design approximation algorithms with provable performance guarantees. We demonstrate their usefulness in four tasks: (i) to generate a summary of a book which is compared to a human summary, (ii) to generate

exemplars for the classic MNIST data set, (iii) to generate exemplars that can be used to identify people and (iv) to perform instance transfer learning. Our approach is based on classic computations (e.g., minimum set cover) but the combination of the methods is novel. This has the advantage of being able to leverage known results and implementations of these classic algorithms; see code in the following repository:

www.cs.ucdavis.edu/~davidson/SCCE-DMKD-main.zip

This has other advantages such as ease of parallel implementation. Like most ML methods, our methods also need parameter tuning. Most clustering algorithms need to tune k (the number of clusters) and our method adds another parameter ϵ (the coverage of an exemplar). The relationship between ϵ and the number of exemplars allows for a natural trade off between the complexity of the explanation and cluster compactness as per our bounds. If the data to be clustered is human interpretable, then other methods of explanation are also suitable [**115**, **116**] but exemplars are a natural and pragmatic way to explain complex data.

CHAPTER 5

Identification and Uses of Deep Learning Backbones via Pattern Mining

Abstract Deep learning is extensively used in many areas of data mining as a black-box method with impressive results. However, understanding the core mechanism of how deep learning makes predictions is a relatively understudied problem. Here we explore the notion of identifying a backbone of deep learning for a given group of instances. A group here can be instances of the same class or even misclassified instances of the same class. We view each instance for a given group as activating a subset of neurons and attempt to find a subgraph of neurons associated with a given concept/group. We formulate this problem as a set cover style problem and show it is intractable and presents a highly constrained integer linear programming (ILP) formulation. As an alternative, we explore a coverage-based heuristic approach related to pattern mining, and show it converges to a Pareto equilibrium point of the ILP formulation. Experimentally we explore these backbones to identify mistakes and improve performance, explanation, and visualization. We demonstrate application-based results using several challenging data sets, including Bird Audio Detection (BAD) Challenge and Labeled Faces in the Wild (LFW), as well as the classic MNIST data.

5.1. Introduction

NOTE: A previous version of this paper is published in SIAM SDM24 [[38, 138]] with co-author Ian Davidson

As models are deployed to tasks traditionally only trusted to humans, understanding a model's behavior is often required. This is particularly true for methods such as deep learning (DL), as their decision-making mechanisms are inherently opaque. Existing work in explainable artificial intelligence (XAI) provides interpretability by explaining a prediction decision on a single instance. While this provides some insight into a particular prediction, it does not demystify the more general decision-making process of the learner. Further, such explanation mechanisms also suffer from an

TABLE 5.1. Taxonomy of XAI into three categories, each with distinct goals and definitions of interpretability.

Category	Interpretability Defintion	Examples
Explaining Model Decisions	Justify a model's action on a particular instance	LIME [16] Counterfactual Explanations [22]
Creating Interpretable Models	Create an inherently interpretable model OR distill an opaque model into an interpretable model	Distilling Networks into Decision Trees [6]
Investigating Model Mechanisms	Provide deeper understanding of how the model processes instance	Feature Visualization [13] Ours

overreliance on the input space. Such explanations are convenient for interpretable input spaces, such as images or text, but may be useless for data with uninterpretable feature spaces, such as embeddings or audio spectrograms as we study in Section 5.7.

In this paper, we explore the area of creating backbones of a deep learner. These backbones can be used for a variety of tasks including identifying mistakes, improving prediction, and global explanation.

Core Idea. A core insight is that any instance activates a subset of neurons in the network. Hence, a concept backbone is a subgraph of hidden units that frequently co-activate for a subset of instances associated with a concept such as a group of instances of a class incorrectly predicted, or some other phenomenon we wish to explain. We can find a collection of concept backbones which are for different concepts and are distinct/different from each other. We refer to this as a collective backbone.

For example, given a network meant to distinguish dogs from cats, one can find a concept backbone for the concept of mispredicting dogs as cats in order to identify future mispredictions and another for correctly predicting dogs to better understand the model's decision-making process. Further, exploiting the distinction between these two provides a basis for accomplishing more complex tasks such as correcting mistakes (Section 5.6).

Our approach is flexible enough to answer a variety of questions. We demonstrate our work on three domains: Labeled Faces in the Wild (LFW) [139], the Bird Audio Detection Challenge

(BAD) [140], and MNIST for visual explanations. In the more challenging LFW dataset, we demonstrate the robustness of our method even when faced with small data, 12-way classification, and class imbalance. We show the versatility of our method by also showing its utility on the non-image datasets of the BAD challenge. Our backbones have high coverage distinctness (the subgraph to cover minimally covers instances from other categories).

We make the following contributions with the last point being important, as justifying the utility of an explanation is critical.

- We present the backbone identification problem for supervised prediction as a coverage problem (See Problem Definition and Formulation), formulate it as an ILP, and prove intractability (See Theorem 1).
- We provide a heuristic algorithm to find a completely connected subgraph covering many instances for the novel concept of a concept-level backbone. A collection of these backbones can explain an entire model (See Section 5.2).
- We prove that this algorithm will produce a solution that is Pareto optimal to the problem in respect to the maximizing problem objective and minimizing relaxation. (See Theorem 2)
- We explore sixteen different networks in three domains (See Section 5.7). Specifically:
  - We apply feature visualization to create explanations from our backbones for the MNIST dataset.
  - We use our backbones to identify mispredictions with high confidence in the LFW and BAD datasets.
  - We use multiple backbones to correct those mispredictions to a great deal of success in the BAD networks.

The paper is organized as follows: we first discuss the core problem and show its intractability, after which we describe our approach. We design and complete our experiments next and then conclude.

[1]

[1] In the interest of space, theorems, proofs, and algorithm 2 are provided in appendix REF

5.2. Overview of Our Approach

Backbone Desiderata. The output of our approach is a concise subgraph of the deep learner that activates with a particular concept/group. We begin by defining the characteristics of a good collective backbone and then a high-level problem formulation.

Every concept-level backbone must:

(1) Describe all the members of the concept

(2) Be distinct from all opposing concepts

(3) Be concise in terms of size

To exemplify our reasoning, consider the following explanations of dogs. One explanation may be "A domesticated four-legged animal with a tail". This is unsatisfactory as this could describe other animals. One can tailor this response to exclude other animals by including details such as "a long snout", however this may exclude some dogs such as pugs or bulldogs. Finally, consider the response by the Oxford Languages Dictionary: "A domesticated carnivorous mammal that typically has a long snout, an acute sense of smell, nonretractable claws, and a barking, howling, or whining voice" [141]. While this is specific and inclusive, in certain contexts, the eloquence of a smaller explanation may be desirable. In the same sense, backbones must be descriptive, exclusive, and ideally concise.

Through this example, one can see that these criteria are at odds with each other. To make an explanation general enough to apply to all members, one may need to sacrifice conciseness. To ensure that the explanation is distinct from other related concepts, one may need to make generalizations that exclude members of the group. This is why backbone discovery naturally lends itself to an optimization setup.

Concept-Level backbone as Finding a Minimal Graph. We view each instance x_k as activating a subset N_k of the model's hidden layer neurons by creating an activation vector of each's nodes influence with each component corresponding to a hidden neuron in the network. *Influence* is calculated as the absolute value of the neuron's activation times the sum of the absolute value of weights. We create a set of node activations $C_i = \{N_1 \dots N_m\}$ for the i^{th} concept for all m instances associated with this concept. Naturally, these vectors can be viewed as graphs in the network or transactions. We then investigate $C_1 \dots C_n$ to understand which neurons are quintessential for the classification of concept i. The goal of a CL-backbone is to find a subgraph of the network such

that the nodes in the subgraph are connected and cover many instances in the concept. This is a challenging problem as the naive approach of taking the union or intersection of the transactions for a given concept yields only trivial backbones (see Figure 5.1).

Below we formalize the problem for a single backbone, and later expand it to a collection of backbones:

The Concept Summarization Problem. Given a set of graphs $G_1 \dots G_n$ of the DL node influences for n instances, find a backbone (subgraph) G^* such that $\forall i$:

- **Complete Coverage** $G^* \subset G_i$
- **Connected** G^* is a connected graph and
- **Conciseness** $|G^*|$ is minimal.

The extension to the Collective backbone problem is then to find multiple CL-summaries $(G_1^* \dots G_k^*)$ with the additional requirement of **Distinctness** from each other $(G_1^* \cap G_2^* \cap \dots \cap G_k^* = \emptyset$ for k concepts). This problem can be easily translated to an ILP, however we prove this to be intractable in Theorem 1, and in practice often infeasible to satisfy. Even relaxations of this ILP are extremely computationally expensive. Instead, we design a coverage-based approach to efficiently find such subgraphs which we prove produces an optimal result to the original formulation of this problem.

5.3. Problem Definition and ILP Formulation

Let X be a set of n data points which, for clarity, are all instances in the same concept (e.g. all incorrectly predicted instances of the same class). Let M be a learned DL model consisting of fully connected nodes R where $R_{l,j}$ is the j^{th} hidden node at layer l. Further, let W be the weights in the DL with W_{n_1,n_2} being the weight connecting node n_1 to n_2. There is no need for M to be trained from X, but this is the case in our experiments. Further, let x_k be the k^{th} instance of the data which activates the subset of nodes N_k, that is $N_k \subset R$. We describe the requirements for activation later. Then X has an analogous representation $N = \{N_1 \dots N_n\}$ which is a set of subsets of node activations that can be represented as a binary $n \times |R|$ table, T, with the entry $T_{k,j} = 1$ representing that instance k activates node j. This is naturally a transaction dataset with the items being node activation. The threshold associated with node activation is application-specific and in

our experiments we threshold so that the r most influential neurons are used. We lift the above notation to any number of categories by using the superscript i.

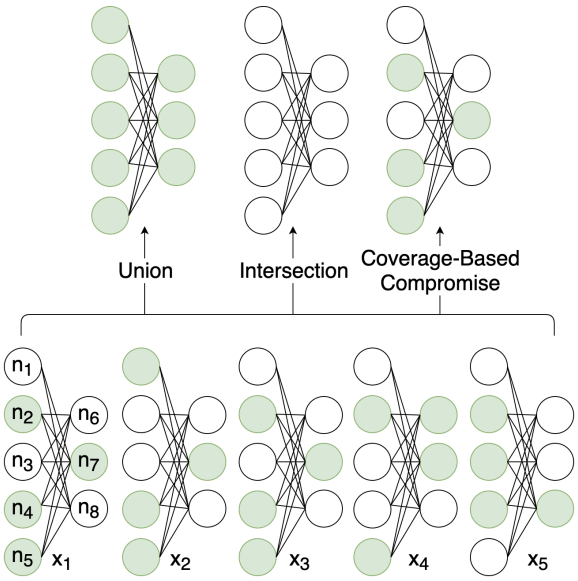


FIGURE 5.1. Potential issues with taking the union and intersection of activation vectors. In this example, there are eight neurons in the network and five instances in the concept. Neurons n_2, n_4, n_5, and n_8 form the clearest summaries, occurring in 80% of the instances and the other neurons in only 20%. The intersection is empty since it requires neurons to be present in all instances, and the union is the whole network since it requires neurons to be present in only once.

Definitions. Before discussing the ILP using notation, we provide definitions for some of the concepts we hope to achieve. In the strict ILP, the backbone must cover all of the instances. That is, the backbone must reflect the activation of all instances $x_i \in X$. This is referred to as **complete coverage**. To create a collective backbone, we also enforce **orthogonality**, in that no two backbones should share a common neuron. In the relaxed ILP, these constraints are no longer strict, and instead, we use the terms **coverage** and **diversity** to refer to the idea that the backbone should cover most, but not necessarily all, instances, and that there should be minimal overlap between backbones.

Problem Statements. Our problem is to find a subset of nodes R^i_* which explains all instances in concept i. A naive version of the problem is to find the largest (hence most descriptive) backbone, which will be equivalent to taking the union of N^i overall i, that is, $R^i_* = \cup_j N^i_j$. However, this risks creating a huge network, and likely produces high overlap between CL-Explanations. Similarly, the

intersection over N^i ($R^i_* = \cap_j N^i_j$) is likely to yield a very small subset of nodes unlikely to form a connected subgraph.

Instead, we model this discovery problem as a set cover style problem. First, we describe the formulation for a single-concept backbone and then extend it to the collective backbone by adding an orthogonality requirement.

$$argmin_{R_*} \sum_i |R^i_*| \tag{5.1}$$

$$s.t.\ R^i_* \subset N^i_j\ \forall i,j \qquad \texttt{Complete Coverage} \tag{5.2}$$

$$s.t.\ R^i_* \cap R^j_* = \emptyset\ \forall i,j,\ i \neq j \qquad \texttt{Orthogonality} \tag{5.3}$$

(5.4)

However, this will return a subset of nodes (R^i_*) for each class i, each of which may not define a connected sub-network, that is, they may not contain a node at each layer in the DL, and there need not even be any connections between the nodes (i.e., non-zero weights). Hence we require any sub-network we find as possessing two properties: i) *Layer Inclusion:* There is at least one node in R^* for each layer in the original network and ii) *Connectivity:* A path using non-zero edge weights between every node in R^i_* exists. For simplicity, we define $C_{R^i_*}(j,k)$ as being the multiplication of the absolute value of the weights that connect nodes j and k using only nodes in R^i_*. Hence the problem we attempt to solve in this paper is given below:

The Connected Concept-Level Backbone Problem.

$$argmin_{R_*} \sum_i |R^i_*| \tag{5.5}$$

$$s.t.\ R^i_* \subset N^i_j\ \forall i,j \qquad \texttt{Complete Coverage} \tag{5.6}$$

$$s.t.\ R^i_* \cap R^j_* = \emptyset\ \forall i,j,\ i \neq j \qquad \texttt{Orthogonality} \tag{5.7}$$

$$s.t.\ \exists n \in R^i_* : n \in R_j\ \forall j\ \forall i \qquad \texttt{Layer Inclusion} \tag{5.8}$$

$$s.t.\ \exists C_{R^i_*}(j,k) > 0\ \forall j,k \in R^i_*\ \forall i \qquad \texttt{Connectivity} \tag{5.9}$$

A proof for intractability of this problem is provided in Theorem 1. Replacing the first two constraints with relaxations may be a solution: not all instances need be covered/explained, but instead, δ_i can be forgotten for concept i, and its description has up to γ_i overlapping nodes with other descriptions.

The Relaxed Connected Concept-Level Backbone Problem.

$$argmin_{R_*} \sum_i |R_*^i| \tag{5.10}$$

$$s.t.\ (N_j^i - R_*^i) \geq \delta^i\ \forall i, j \qquad \texttt{Coverage} \tag{5.11}$$

$$s.t.\ (N_j^i - R_*^i * R_*^j) \geq \gamma^i\ \forall i, j,\ i \neq j \qquad \texttt{Diversity} \tag{5.12}$$

$$s.t.\ \sum \delta < p_1 \qquad \texttt{Coverage Relaxation} \tag{5.13}$$

$$s.t.\ \sum \gamma < p_2 \qquad \texttt{Support Relaxation} \tag{5.14}$$

$$s.t.\ \exists n \in R_*^i : n \in R_j\ \forall j\ \forall i \qquad \texttt{Layer Inclusion} \tag{5.15}$$

$$s.t.\ \exists C_{R_*^i}(j,k) > 0\ \forall j, k \in R_*^i\ \forall i \qquad \texttt{Connectivity} \tag{5.16}$$

Where p_1 and p_2 are the maximum number of instances that can be forgotten and the number of overlapping nodes, respectively.

5.4. Approach

In this section, we discuss our heuristic-based solution to the *Connected Concept-Level backbone Problem*. In later sections, we mathematically prove and empirically demonstrate that this algorithm is guaranteed to provide an optimal result. We accomplish this through two simple but efficient algorithms: Find Max Minsup (FMM) (Algorithm 4), which finds the connected layer-inclusive subgraph with the highest support, and F-Score Thresholding (Algorithm 2), which iteratively adds new neurons with the greatest support that either form or adds to a complete graph, to the backbone which maximizes our heuristic.

FMM is a frequent subgraph mining algorithm that finds the most frequent subgraph that meets the fully connected and inclusive layer constraints. Frequent subgraph mining requires a minimum support threshold to be specified [142] [143], however there is no way to immediately know what value of minsup will generate a frequent subgraph satisfying the constraints of the problem. To

deal with this, minsup is first set to 100% of the data (the intersection of the transactions) and decremented by a single instance each iteration until a complete (layer inclusive and connected) graph is created, returning the value of support of that subgraph. While it may sound appealing to perform binary search to find this value, frequent pattern mining algorithms tend to grow exponentially in computational complexity as minsup becomes smaller [144], so the method presented is more efficient. Furthermore, since minsup can be decremented to zero, and since each transaction has at least one connected neuron from each layer, we are guaranteed to find the single subgraph with the greatest support. We dub this subgraph the backbone.

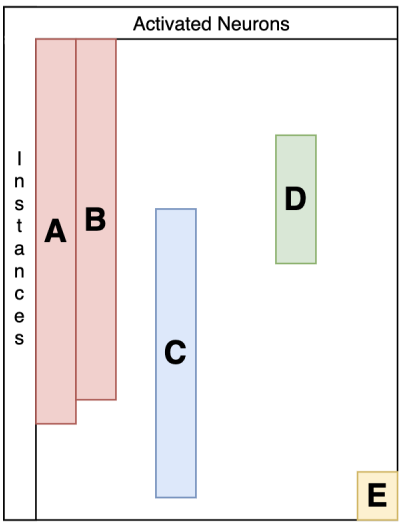


FIGURE 5.2. A visualization of the matrix of node activations N as a series of transactions with columns as different neurons and rows as instances. Color corresponds to patterns, and groups of neurons are labeled. FMM only finds group A, but ignores everything else. F-Score thresholding allows groups B and C to be included in the backbone despite having lower support than max minsup. Groups D and E have much lower support, so they will not be included.

The backbone only includes the most frequent complete graph, however, and will ignore patterns that are nearly as frequent (see Figure 5.2). F-Score Thresholding finds a Pareto optimal solution regarding the ILP's objective and minimizing the relaxations. That is, the graph returned cannot simultaneously have greater coverage, diversity, and/or be smaller (see Theorem 2).

We accomplish this by viewing the backbone as a predictive model for which neurons will appear in a transaction and iteratively adding the next most frequent pattern until the change in F-score after one iteration is negative. To calculate an F-Score, we define the true positives as the simultaneous occurrence of a neuron in the backbone and transaction, a false positive as a neuron

occurring in the backbone, but not the transaction, and a false negative as a neuron occurring in a transaction contains a neuron. Precision and recall are determined in traditional ways, and F-Score is the harmonic mean between the two. It is important to recognize that maximizing recall maximizes the instances covered, whereas precision acts as a check, penalizing the heuristic for adding infrequent neurons. In order to distinguish between the patterns, a weight is given to each pattern in the backbone equal to that pattern's support divided by max minsup.

Algorithm 4 Input: set of activation vectors N
Output: Value of minsup that produces a graph with the highest support

$s \leftarrow 1$ // Minimum Support
$d \leftarrow 1/\text{len}(N)$ // Support decrement
$subgraph \leftarrow \text{patternMining}(N, s)$
while $\neg\text{completeGraph}(subgraph)$ **do**
    $s \leftarrow s - d$
    $subgraph \leftarrow \text{patternMining}(N, s)$
return s

Algorithm 5 Input: Max-minsup from Algorithm 4, activation vectors N, minimum coverage λ (optional)
Output: Weighted graph.

$maxF \leftarrow -1$ ▷ Maximum F Score
$sum \leftarrow \emptyset$ ▷ Backbone
$d \leftarrow 1/\text{len}(N)$
$s \leftarrow \text{maxMinsup}$
while True **do**
    $potentialGraphs \leftarrow \text{patternMining}(N, \text{minsup})$
    for graph in potentialGraphs **do**
        if completeGraph(graph) **then**
            $sum \cup$ graph
    $FScore \leftarrow \text{getFScore}(N, \text{graphs})$
    if $FScore <$ maxF & $\text{cov}(sum, N) \geq \lambda$ **then**
        break
    maxF $\leftarrow FScore$
    $s \leftarrow s - d$
$R \leftarrow \emptyset$
for graph in graphs **do**
    $R \leftarrow R \cup \text{graph}, (\text{support}(N, \text{graph}))$
return R

5.5. Models and Datasets

In order to demonstrate our technique's invariance to domain and utility on very different types of networks, we conducted experiments using sixteen different networks in three different domains, raw image data from the MNIST dataset, audio data from the Bird Audio Detection Challenge, and embeddings generated from Facenet of faces from the Labeled Faces in the Wild (LFW) dataset. The Section 5.8 provides a description of the datasets, model architectures, and why each of these networks is interesting. Each backbone is referred to by the dataset used to create them, with a superscript + indicates that the backbone for correctly predicted instances of a given class and - for incorrectly predicted instances. Unless + or - is given, it is assumed to be the + backbone.[2]

5.6. Experimental Design

These results are based on summaries generated from all folds for each network. Details on the datasets, networks, and cross-fold validation are provided in Section 5.8.

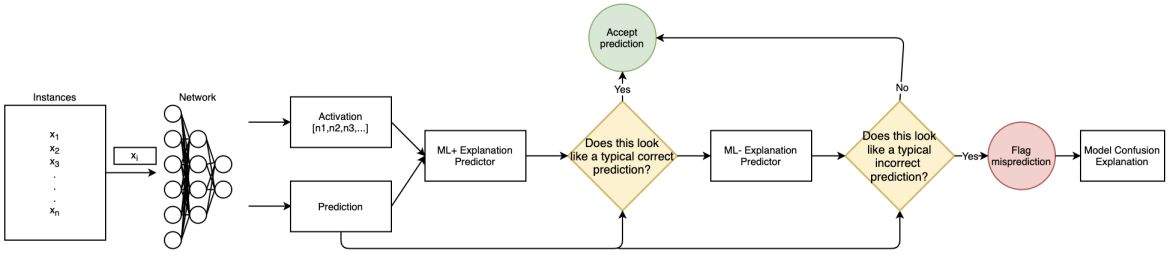


FIGURE 5.3. Flow diagram of the process of flagging mispredictions and correcting them using the collective backbone and the prediction of the network.

Heuristic Approach Compared to ILP. Before demonstrating interpretability, we compare the heuristic-based solution to that of the ILP. To compare the solutions of the two approaches, we see how two metrics, coverage and overlap, change as new subgraphs are added to the backbone and use relaxed formulations of the ILP as baselines. We say that an instance x is covered by a backbone C_i if some complete connected subgraph exists in the activation vector of x that also exists in C_i. Overlap between two or more summaries is the number of neurons that both have in common with each other divided by the size of the summaries. In the case of the BAD network, the ILP is relaxed until solutions can be generated in 24 hours or less, and in the LFW network,

[2]Trained models, datasets, results, and intermediary results are included at https://github.com/MLivanos/backbonesSDM24

they are relaxed until Pareto optimality. Ideally, we would see coverage monotonically increase and diversity minimally decrease as new patterns are added until it reaches a desirable point on the Pareto front.

Backbone as a Predictive Device. If a backbone of a concept (such as a class) is robust enough, one should identify the concept when they encounter it. Here, we compare the activation vector for a given instance to each of the CL-summaries, and assign a prediction to the most similar one. Since these are summaries of the concept, we do not expect the accuracy of the backbone to be as high as that of the model, however we do expect a good backbone to have comparable results.

Predicting Mispredictions and Correcting Them. We create a pipeline for detecting and correcting mispredictions of the model, summarized in Figure 5.3. As opposed to the previous experiment, we consider both the activation vector for a given instance and the network's prediction of that instance. Using the same method described in the above experiment, we compare the activation vector to the set of correctly predicted CL models, asking the question: "Does this look like a typical correct prediction of the predicted class?" If the prediction of the network and that of the summaries differ, we repeat the process on the incorrectly predicted instances, asking the question "Does this look like a typical misprediction of this class?". If the answer to the first question is no and the second yes, the prediction is assumed a misprediction.

After being flagged, an alternative prediction is provided. For binary classification, this is trivial, as the prediction is simply swapped to the other class. In the case of multi-class classification, a model confusion backbone is provided, a CL backbone in which the concept is "class x being predicted as class y" for all combinations of mispredictions, and this backbone is used to determine the alternative prediction.

Subgraph Visualization. We perform feature visualization to create human interpretable explanations to find virtual instances in the input space that most activate the subgraph returned from our method. We use particle swarm optimization (PSO) [60] to minimize the euclidean distance between the normalized activation vector and the backbone returned from our method. The result is a virtual instance whose activation is high in the neurons of the backbone but not any other neurons. Because it would be difficult to leverage the cost function gradient to optimize activation for neurons on different layers, PSO was chosen as the algorithm for this task since it is a gradient-free optimization algorithm.

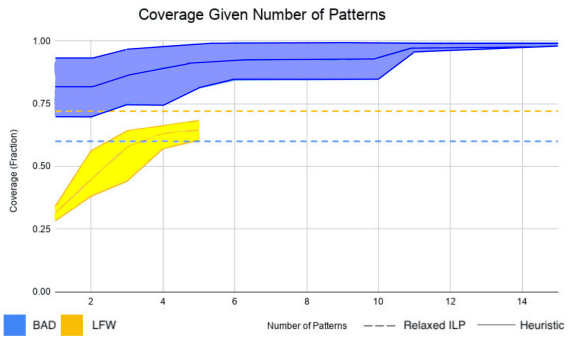
In order to focus the algorithm and produce crisper explanations, we create a pixel whitelist of the top 40 percent of pixels found in each class. While this reduces and focuses the input space, the optimization stays the same.

5.7. Experiments

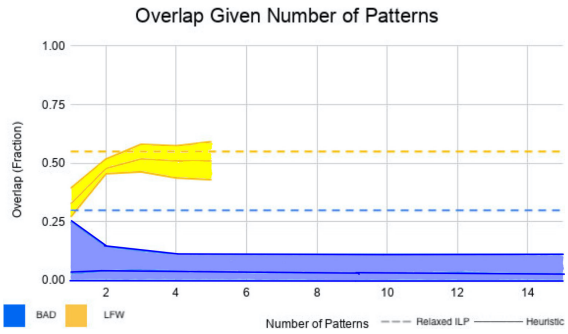
Heuristic Approach vs ILP. Theorem 2 proves that our algorithm finds a non-trivial Pareto-optimal solution with respect to minimizing the objective of the problem and the two relaxations, however it is not immediately discernible where on the Pareto-front the solution will lie. Further, while the ILP formulation of the problem is proven intractable in Theorem 1, we empirically examine the speedup of the heuristic solution. In this experiment, we create 15 explanations via our approach and compare them to that of an ILP to demonstrate the viability and efficiency of our algorithm.

In the BAD network, the ILP must be relaxed greatly in order to return a solution within 24 hours. As shown in Figure 5.4, the heuristic solution achieves an initial 80% coverage, and after adding 14 additional patterns, achieves nearly 100%, compared to the ILP's 60%. Overlap starts at about 2.8% and only increases to 3% at the end, compared to the ILP's 30%. Not only does our method substantially outperform the relaxed ILP on both metrics, it does so in an average of 12 minutes compared to the ILP's 24 hours, speeding up the process by a factor of 120.

The LFW summaries also drastically increase in coverage as patterns are added to the backbone. In this case, we see that the solution returned by the heuristic approach sacrifices some accuracy for diversity. Five patterns are added before the algorithm terminates; increasing coverage increases from 31.1% to 64.5%, compared to the ILP's 72%. While there is a jump in overlap from the first pattern added to the second, afterward overlap does not significantly increase, ending at 51% compared to the ILPs 55%. Moreover, it takes 10 minutes, compared to the ILP's required 30 minutes.



(a) Coverage (vertical axis) against number of patterns added (x-axis).(higher is better)



(b) Overlap (vertical axis) against number of patterns added (x-axis). (lower is better)

FIGURE 5.4. Quantifying coverage and overlap difference between the relaxed ILP and heuristic. For both datasets, the top line represents the maximum (across folds) for that metric, the middle the median, and the lower the minimum. Coverage increases over iterations while overlap minimally increases.

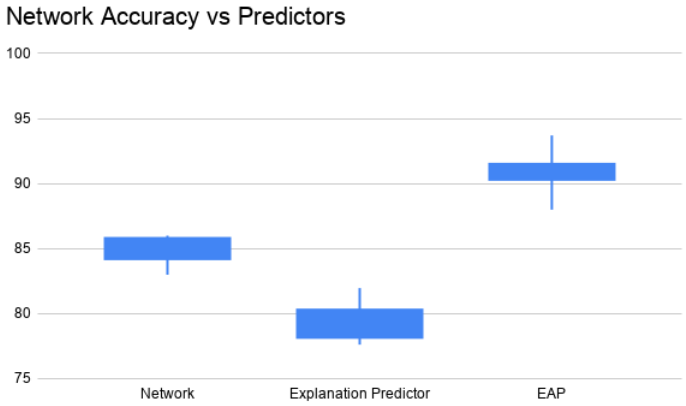


FIGURE 5.5. The network, backbone as a predictive model, and the explanation augmented predictor accuracy on BAD test data. When used as a predictive device, the backbone underperforms the network, as expected, however when one considers both the backbone and the output of the network, as one does in the EAP, accuracy is increased significantly.

Predictive Device. We demonstrate the quality of our explanations by showing that they alone can be used for classification and achieve similar results to that of the networks they explain. If the explanation of the model's behavior can be used in this way, then the explanation is good and faithful to the model.

In both the LFW and the BAD summaries, predictive capabilities had lower but comparable accuracy than their respective models. The median accuracy for the LFW summaries was 45%

compared to the median model's 60%, and the median accuracy for BAD summaries was 78.5% compared to the model's 85.2%. Interestingly, despite having lower accuracy, the predictive device can correctly classify instances that the model could not. This observation was the impetus for creating the explanation augmented predictor.

Explanation Augmented Prediction. To demonstrate the practical utility of our explanations, we use them to identify when the models tend to fail and correct their predictions.

In the BAD Challenge dataset, we correctly flag nearly two-thirds of incorrectly predicted instances while only incorrectly flagging 5% of correctly predicted instances as mispredictions. This allowed us to create a model of greater predictive power than the original network, elevating the median accuracy from 85.2% to 91.3%. In addition, all ten folds exhibited an increase in accuracy ranging from 5 to 7.3%. See Figure 5.5.

In the LFW dataset, we correctly identify 21.4% of incorrectly predicted instances, and incorrectly flag 12.1% of correctly predicted instances. While 33% of mispredictions could be corrected using the model confusion explanation, augmenting these networks would, on average, have lower accuracy than the model on its own. While the LFW explanations can identify when errors occur, it cannot reliably correct them likely due to the small data nature of the problem.

Subgraph Visualization. The neuron visualization technique provides, in human interpretable terms, the semantic meaning of the graphs returned in Algorithms 4 & 5. The results of this approach are in Figure 5.6a. This provides insight and validity to the backbone, allowing the user to understand how it is activated. Figure 5.6b shows the maximization of different subgraphs in the backbone. This allows the user to see how the component subgraphs capture different parts of the concept of each digit, such as the tail vs. the head of the five.

(a) Visualization of the semantic meaning for each concept (digit) in the collective for the MNIST dataset, accomplished via activation maximization)

(b) Visualization of the semantic meaning of each component subgraph of the CLE of the digit 5. Different parts of the digit are activated by different subgraphs. For example, ii. focuses on the tail of the 5, whereas v. focuses on the head, and i. on the middle section. Under each instance is the support that subgraph has over the instances.

FIGURE 5.6. Caption for the entire figure

5.8. Reproducibility Details: Model Architecture and Dataset Selection

FaceNet Embeddings From Labeled Faces In The Wild. FaceNet was created by Schroff et al 2015 to generate embeddings that have small Euclidian distance between two tight-cropped faces of the same person, but have larger Euclidian distance between different people [28]. Scroff et al show that embeddings have smaller distances comparing the same face from different angles and lighting than a different face in the same angle and lighting.

We use a subset of the Labeled Faces In The Wild dataset to create a single 12-way classification network trained on the embeddings for each individual. Those individuals were those who has at least 50 images in the dataset. The dataset has significant class imbalance, with the least represented individuals (Serena Williams and Jacques Chirac) having only 53 images, and the most well-represented person (George W. Bush) having over 500. This network is composed of five fully connected hidden layers with 80, 60, 40, 30, and 20 neurons respectively. We train five networks using cross-fold validation with a class-balanced test set and median accuracy of 60% for the 12-way classification task.

Since our approach requires at least one neuron per layer in the backbone, the network's wide design means that we will have long, spanning summaries. Further, the class imbalance in training, low network accuracy, and lack of available data will pose challenges that our technique will need to overcome. These challenges make explaining this network the hardest task.

Bulbul: Bird Audio Detection Challenge. Bulbul was developed by Grill et al 2017 as part of the Bird Audio Detection (BAD) challenge. Mel spectrograms generated from raw WAV files are given as input, most of them 10 seconds long. The training data comes from multiple sources, each from different regions of the world, different recording equipment, and different class balance. [140] Bulbul was the winning network of the challenge in 2018, achieving an area under the curve (AUC) of 88.7% [145]. Here, we recreate bulbul using 10-fold cross validation and achieve a median accuracy of 84% on the validation set. Two notable difference between the networks that we trained and that of Grill et al is that, at the time of the experiments, The Machine Listening Lab (the entity in charge of the BAD Challenge) has not published the labels of their testing data, so we use a subset of the training dataset (that was not used for training of our network) as testing data. Also, our model was trained on a fixed number of epochs, while Grill et al use a variable training scheme. We also report our findings in terms of accuracy, rather than AUC as Grill et

al did, and demonstrate that, using backbone to augment prediction, we can significantly improve accuracy. Due to these differences, we do not claim superiority over Grill et al's method, but we do demonstrate that our method can surpass a similar network with the same architecture and training data on the metric of accuracy.

Bulbul is a much shallower network, with only two dense layers with 256 and 32 neurons respectively. This will create smaller summaries than the LFW summaries. Further, this network is for binary classification and is trained on large datasets (over 10,000 training instances). Due to these factors, we suspect that summaries for this network will work well with our experiments and yield us better results. Finally, this network was chosen to showcase our technique's result on a network of high domain importance, as this was the top model of the 2018 BAD challenge.

MNIST The MNIST digit recognition network is trained directly from the MNIST training set. It is composed of two convolutional layers, with 32 and 64 channels, respectively, each with a 3x3 sliding window. Following each of the convolutional layers is a max pooling layer with a 2x2 sliding window, connected to two fully connected layers of size 64 and 32 neurons.

5.9. Related Work

In the preceding sections, we have discussed our approach and demonstrated its utility on complex XAI tasks. Here, we discuss other approaches and some of their deficiencies which our approach has overcome. As opposed to our work, which provides category and model-level explanations grounded in model architecture, most existing XAI methods explain a model's behavior on specific instances and/or ground their explanation in input space. In this section, we highlight the need for our particular form of explanation in contrast to existing methods.

Many XAI methods provide local interpretability, that is, an explanation for a single instance [146]. Popular techniques that provide this incredibility include those that isolate superpixels of an instance [14], [147], or counterfactual explanations [148] which generate virtual instances to explain why a different action was not taken. This provides limited insight to the future behavior of the model because it only speaks for the instance that it is explaining and with no guarantee that the model will behave the same way for future instances.

The input space is a natural choice for presenting an explanation, as it is often inherently interpretable, facilitating human perceptive explanation [149]. However, numerous works towards in

adversarial attacks demonstrate how engineered, imperceptible changes are drastically alter a network's prediction [150], [151]. Grounding an explanation in the architecture, by contrast, focuses exclusively on how the network processes information, allowing a more complete examination of the model's behavior. One example of this deficiency exists in the field of feature visualization. This model-level explanation technique grounds the semantic meaning of layers or neurons in the input space [152]. Researchers note that semantically different images can achieve similar levels of activation [153], demonstrating the volatile nature of input space as the basis for explanation.

5.10. Conclusion

We formulate the problem of discovering concept and collective backbones as subgraphs of a deep learner, prove that the ILP formulation of this problem is intractable and expensive even with relaxations. We propose a heuristic-based approach via frequent subgraph mining techniques and prove that this method returns a Pareto optimal result with respect to maximizing the problem's objectives and minimizing relaxations and does so at a fraction of the runtime.

These summaries provide a basis for completing complex XAI tasks that existing methods cannot. Our approach can determine patterns in model failure which can be used to determine mispredictions, patterns in model success, and use the combination of those two to correct mispredictions. For example, our method succeeded in boosting the performance of the BAD network and could successfully identify errors in the LFW network, although it could not correct them. This indicates that our method performs best on high-performance and binary classification networks trained; however, further investigation is required to understand which of these factors is most pressing.

This work differs from most XAI research as it presents model level summaries grounded in hidden-neuron space that can be used in ways to improve the trustworthiness of a model and provide greater insight into how the learner makes decisions.

CHAPTER 6

The Intersectional Unfairness Paradox: An Empirical Investigation Of Intersectional Fairness

Abstract As machine learning is deployed in social contexts, addressing the biases an algorithm can learn during training is increasingly important. Fairness in machine learning seeks to solve this problem by training algorithms to be both performant and equitable across protected statuses (eg gender or race). The majority of these algorithms tend only to consider these protected statuses individually and do not consider or measure how machine learning models can be unfair to intersections of these groups (e.g. combinations of particular genders and races). While there has been some promising preliminary work toward addressing this deficiency, in this paper, we examine the effect of fairness interventions on both individual protected status variables (PSV) and intersectional fairness on 8 of the most popular datasets used in fairness research using multiple fairness metrics. We demonstrate the counterintuitive result that making an algorithm fair with respect to individual PSVs tends to decrease intersectional unfairness involving those same PSVs.

6.1. Introduction

While the decisions of a machine may outwardly appear to be unbiased, research has consistently shown that bias can be embedded into a model through its training data and/or learning. To address these concerns, research in fair machine learning has sought to create models that actively combat bias to produce more equitable outcomes. The vast majority of these works, however, only seek to make a model fair with respect to protected status variables individually [154]. Even when fairness is encoded for multiple protected statuses it is encoded separately. For example, one may try to train a model to be fair with respect to gender, race, or both gender and race, but ignore the intersections between them. This is in conflict with the decades of social science research into intersectionality, the idea the combination of different dimensions of identity can create new forms of bias that differ from the sum of their parts [155].

To illustrate how unfairness can exist for intersections of PSVs even if the model is fair to multiple PSVs simultaneously, we create an illustrative example in Figure 6.1. Before any fairness intervention, the original model (bottom) is unfair by any fairness definition, with a bias favoring triangles over squares and orange over blue. The upper left model is made fair with respect to shape, though bias for color persists. The upper center model is made fair with respect to color, though bias for shape persists. The upper right has been made fair with respect to both color and shape individually, but intersections of color and shape are still treated unfairly - blue triangles and orange squares now shoulder the burden of making the model appear fair even though the model is clearly discriminatory to certain groups. While this model is considered perfectly fair in the eyes of certain fairness algorithms, it is clearly not fair when intersectionality is considered.

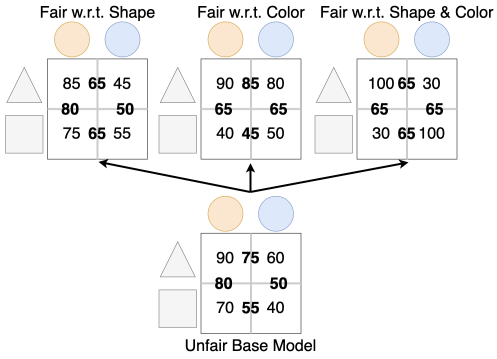


FIGURE 6.1. A demonstration of how a machine's output can be made fair with respect to a single protected status variable (top, left and center) or multiple individual PSVs separately (top right) but be unfair to intersections of those PSVs (Here, shape and color) We assume for simplicity that all intersections are equally represented in the data. The number inside each box represents some fairness metric (eg accuracy, false positive rate, positive class representation, etc) of a particular intersection, and the bold numbers in between boxes are those metrics for individual PSVs.

Our research empirically investigates how existing fairness algorithms that do not account for the intersection of multiple PSVs, what we call intersectional-unaware fairness, affect fairness metrics on the individual and intersectional basis. Our contributions are:

- We demonstrate that while one can typically successfully increase fairness with respect to one PSV or multiple PSVs individually, there is no guarantee that intersectional fairness will improve (see Table 6.1).

Law School	Enforcing Fairness w.r.t	Task Acc.	Gender			Race			Inter-section		
			SP	EO	PE	SP	EO	PE	SP	EO	PE
Trained w/o Fairness	None	90.7% ±6.8%	0.021 ±0.017	0.11 ±0.018	0.011 ±0.018	0.057 ±0.0084	0.13 ±0.065	0.58 ±0.35	0.021 ±0.017	0.24 ±0.058	0.30 ±0.21
Fair - Fair Batch	Gender	90.3% ±5.7%	0.009 ± 0.010	0.013 ±0.015	0.0058 ±0.017	0.047 ± 0.0071	0.13 ±0.067	0.36 ±0.42	0.026 ±0.060	0.18 ±0.042	0.35 ±0.22
	Race	88.5% ±7.0%	0.028 ±0.022	0.33 ±0.022	0.019 ±0.026	0.028 ±0.0065	0.092 ±0.052	0.18 ±0.26	0.047 ±0.061	0.31 ±0.057	0.53 ±0.44
	Gender & Race	86.2% ±9.7%	0.011 ±0.024	0.09 ±0.024	0.012 ±0.015	0.022 ±0.0070	0.11 ±0.069	0.32 ± 0.30	0.034 ±0.062	0.30 ±0.041	0.38 ±0.20
Fair - Adversarial [157]	Gender	88.3% ±3.8%	0.0015 ±0.011	0.0011 ±0.015	0.0044 ±0.0047	0.048 ±0.0062	0.09 ±0.056	0.45 ±0.18	0.011 ±0.055	0.19 ±0.048	0.27 ±0.062
	Race	88.3% ±3.4%	0.022 ±0.019	0.22 ±0.019	0.013 ±0.011	0.018 ±0.055	0.058 ±0.055	0.22 ±0.17	0.032 ±0.031	0.35 ±0.14	0.58 ±0.054
	Gender & Race	86.7% ±2.2%	0.0018 ±0.030	0.0072 ±0.0053	0.018 ±0.066	0.027 ±0.081	0.071 ±0.023	0.35 ±0.054	0.023 ±0.035	0.24 ±0.059	0.32 ±0.068

TABLE 6.1. Fairness and accuracy metrics for neural networks trained using the Law School dataset. The protected statuses are gender (treated as binary) and race (five discrete categories). Orange denotes a degradation in performance/fairness, and blue an improvement compared to the baseline, with darker shades indicating greater values. In all columns except "Accuracy", lower numbers are better (ie more fair). This trend exists for other data sets listed in Section 6.2.

- Earlier work shows making algorithms fair to multiple PSVs either separately or in combination is intractable [156].
- We find that existing algorithsm when encoding fairness tend to decrease intersectional fairness.
- We hypothesize why this occurs and gives direction to our future work.

6.2. Approach

To truly understand the extent to which intersectional unaware algorithms impact intersectional fairness, we chose a wide range of datasets, fairness metrics, and algorithms and test all combinations of these factors. Using a recent survey on the datasets used in fair machine learning [158], we use all datasets where intersectional fairness can be extracted (ie all those with two PSVs): KDD Census Income, Bank Marketing, COMPAS Recidivism & Violent Recidivism, Students Math and Portugese, and Law School datasets. A survey on fairness definitions [154] cite 20 distinct fairness measures, we chose the three most popular statistical measures of fairness: statistical parity (SP, Equation 1), equalized odds (EO, Equation 2), and predictive equity (PE, Equation 4). When measuring how close any of these definitions are to being achieved, we take the absolute difference between the right and left sides of the equality. For multiple PSV's, we consider the average difference between all combinations of groups. Intersectional fairness is also measured with respect

to these formulations, with every combination of protected status being considered their own PSV. Finally, we use two fairness algorithms: fair-batches, in which the output of every batch is reorganized so that the predictions are equal in proportion to particular PSV(s), and an adversarial fairness approach presented in [157].

$$P(y' = +|PSV = \alpha) = P(y' = +|PSV = \beta) \quad (1)$$
$$P(y' = y|Y = y, PSV = \alpha) = P(y' = y|Y = y, PSV = \beta) \forall y \in \{+, -\} \quad (2)$$
$$FPR = FP/(FP + TN) \quad (3)$$
$$FPR_{PSV=\alpha} = FPR_{PSV=\beta} \quad (4)$$

Data is segmented into a train and test set via a random 80/20 split. The model is trained with or without fairness-algorithm intervention, and the model's test set accuracy and fairness metrics are recorded. This process is repeated 30 times, and the mean result and standard deviation are reported.

6.3. Results & Conclusion

While one can usually make a learner fairer with respect to a single PSV or multiple PSVs individually, fairness algorithms are not guaranteed, nor tend to, be fair on any other metric. Consider our example for the Law School dataset in Table 6.1. Generally, at the cost of accuracy, a learner can become more fair to a single PSV when fairness is enforced with respect to that PSV or, typically more modestly and at a greater cost to performance, to multiple PSVs. Fairness metrics for intersections were almost always worse across all three metrics despite fairness being encoded for each PSV individually. In the law school example, out of the 18 points of comparison, (six networks and three metrics), 13 had lower intersectional fairness metrics. Out of the 144 total comparisons across all 8 datasets, 142 yielded improved the fairness metrics they were optimized for, although only 32 improvement in intersectional fairness.

The Law School dataset provides two protected statuses, gender (presented as a binary), and race (White, Black, Asian, Hispanic, and Other), with the baseline model showing preference to White men. When the models are made fair to both gender and race simultaneously, the groups most commonly disadvantaged are White men, White Women, Black men, and Asian Men, the latter three groups already disadvantaged in the baseline model. We hypothesize that, as in Figure

6.1, the network favors certain combinations of identities and disfavors others to boosts the fairness metric while minimally impacting the bias relied upon for accuracy.

Fairness exists to combat data biases and make more ethical AI, though the formulations and assumptions of fairness research do not always align with social scientists' theories about how bias occurs. Here, we examine one example of this - a lack of intersectional-awareness encoded into fairness algorithms. We demonstrate that even when fairness algorithms are successful, they do not always ameliorate issues faced by certain groups of people, and metrics that do not consider such groups mask biases. Given these results, we will explore the understudied area of intersectional-aware fairness, and encourage AI researchers to work closer with social scientists who can potentially identify similar issues in the area.

CHAPTER 7

Foundations Of Unfairness in Anomaly Detection - Case Studies in Facial Imaging Data

Abstract

Deep anomaly detection (AD) is perhaps the most controversial of data analytic tasks as it identifies entities that are specifically targeted for further investigation or exclusion. Also controversial is the application of AI to facial data, in particular facial recognition. This work explores the intersection of these two areas to understand two core questions: *Who* these algorithms are being unfair to and equally important *Why*. Recent work has shown that deep AD can be unfair to different groups despite being unsupervised with a recent study showing that for portraits of people: men of color are far more likely to be chosen to be outliers. We study the two main categories of AD algorithms: autoencoder-based and single-class-based which effectively try to compress all the instances and those that can not be easily compressed are deemed to be outliers. We experimentally verify sources of unfairness such as the under-representation of a group (e.g people of color are relatively rare), spurious group features (e.g. men are often photographed with hats) and group labeling noise (e.g. race is subjective). We conjecture that lack of compressibility is the main foundation and the others cause it but experimental results show otherwise and we present a natural hierarchy amongst them.

7.1. Introduction

Anomaly detection (AD) is a central part of data analytics and perhaps the most controversial given that it is employed for high impact applications that identifies individuals for intervention, policing and investigation. It's use is prevalent to identify unusual behavior in finance (transactions) [159, 160], social media (posting and account creation) [161, 162], and government services (medicare claims) [163, 164].

Perhaps one of the most controversial applications of AI is to facial imaging. This is due to our faces being uniquely identifying and personal. Further, the AI's ability to identify us and make

decisions on (without consent) crosses many cultural and legal barriers [165]. Existing work on facial data has focused predominantly on facial recognition, that is, given a large collection of people in a known database, identify if any of them occur in an image. Though legislation and progress has been made towards regulating facial recognition technology [166] other technologies in particular AD involving facial images are starting to emerge which gives rise to new ethical considerations and understanding.

Previous work [167] has just begun to explore the unfairness at the intersection of AD applied to facial imaging data. For example, this previous work showed that applying AD to a collection of celebrity images overwhelmingly showed the anomalies being people of color and males (see Figure 7.1). However, this previous work was mainly focused on making AD algorithms fairer. We recreate their earlier results for not only the one-class AD method and the celebrity image dataset the authors used but also for the popular auto-encoder AD method and a more challenging dataset (Labeled Face In The Wild [139]).

Our experimental section attempts to address the "Who" and "Why" questions. We create a measure of unfairness (anomaly DIR) which measures how over-represented is a protected group (or it's complement) in the anomaly set. We then experimentally investigate who these algorithms are being unfair to and more nuanced questions such as is the same group always being treated unfairly regardless of algorithm. We also explore why an unsupervised algorithm can be biased. We conjecture four main foundations of unfairness, propose metrics to measure them and outline a series of experiments to test a hypothesis on how they are structured.

The contributions of this work as are as follows:

- We study the intersection of anomaly detection and facial imaging data - a topic to our knowledge has not been addressed before focusing on the "Who" and "Why" questions.
- Our experiments addressing the "Who" question show that unfairness is due to an interaction between the dataset and the algorithm.
- We conjecture four main reasons to the "Why" question: i) incompressibility, ii) sample size bias (SSB), iii) spurious feature variance (SFV) within a group and iv) attribute/group labeling noise (ALN).
- We postulate an intuitive structure to our conjectured reasons, show it is not empirically verified, and craft an alternative structure based on the results of our experiments.

We begin by discussing background and related work. We then introduce how we measure unfairness in AD and our four proposed foundations of unfairness. Next, our experimental results addressing the "Who" and "Why" questions are presented after which we discuss and conclude our work.

FIGURE 7.1. Example of AD Being Unfair When Applied to Facial Imaging Data. Reproduced from [167].

7.2. Background and Related Work

Applications of AD to Facial Data. AD algorithms have been used on imaging data for a variety of reasons. Perhaps the most ubiquitous is for data cleaning where anomalies are viewed as being "noise" [168] which are removed and then a downstream supervised algorithm is applied. However, if the AD algorithm is biased this creates an under-representation in the down-stream training tasks of the over-represented group in the outliers.

Another common use of AD is to view the outliers as "signal" and in doing so flag the outliers for extra attention. Examples include using AD to identify facial expressions to recognize emotions [169] such as surprise. However, if the AD is biased towards some groups this will over-predict certain emotions for certain groups. Similarly, AD can be used to identify aggressive behavior [170].

However, if the AD has a bias towards some groups this will incorrectly identify the group as being overly aggressive.

Source of Bias. It has been well established that supervised learning algorithms can have bias due to a variety of reasons. In particular class labeling bias has been extensively studied in the context of the Compas dataset [171]. Even though features (e.g. race) associated with this bias are removed deep learning offers the ability to learn surrogates (e.g. zip code) [172].

The work on fair AD starts in 2020 [173, 174] and has shown that AD algorithms can cause bias. Most work has focused on how to correct unfairness for a certain algorithm. This involves understanding the limitations in the algorithm's computation and then correcting for it. This has been explored for classic density-based methods such as LOF [174] and deep learning methods for autoencoder [175], one class [167] and multi-class deep AD methods. However, despite this earlier body of work, there has been surprisingly little work discussing what produces unfairness in unsupervised anomaly detection.

7.3. Four Reasons for Unfairness And Their Measurement

Here we outline our four premises for unfairness in AD and explain them at a conceptual level using Figure 7.1. We then describe how we measure them.

7.3.1. Incompressability of Data. We begin by discussing how AD methods work in particular what causes an instance to be an outlier. Deep AD methods at their core employ compression either directly or indirectly. Instances that cannot be compressed well are deemed outliers and if a group is unusual in some sense it will be unfairly treated as it will be hard to compress and hence overwhelmingly flagged as an outlier.

To understand this further, we present a common taxonomy of anomaly detection algorithms [176].

Autoencoder for Anomaly Detection. Let ϕ_e be the encoding network which maps the data X into the compressed latent space and ϕ_d be the decoding network which maps the latent representation $\phi_e(X)$ back to the original feature space [177]. Given the network parameters θ_e, θ_d the standard reconstruction objective to train the autoencoder is:

$$argmin_\theta \left(\frac{1}{n} \sum_{i=1}^{n} \|x_i - \phi_{\theta_d}(\phi_{\theta_e}(x_i))\|^2 + R \right) \tag{7.1}$$

The term R denotes the regularization to the encoder and decoder. The anomaly score $s(x)$ for instance x is calculated from the reconstruction error:

$$s(x) = ||x - \phi_{\theta_d}(\phi_{\theta_e}(x))||^2 \tag{7.2}$$

Here clearly an outlier is defined as being an instance that the AE cannot easily compress and hence cannot easily reconstruct [178].

One-Class/Cluster Anomaly Detection Next, consider one class anomaly detection which is still unsupervised. Given the training data of instances $X \in R^{n \times d}$, one class AD method such as the the popular deep SVDD [179] network is trained to map all the n instances close to a fixed center c. Denote function ϕ as a neural network with parameters θ the training objective function is:

$$argmin_\theta \frac{1}{n} \sum_{i=1}^{n} ||\phi_\theta(x_i) - c||^2 + R \tag{7.3}$$

where the term R represents the regularization function. Then the anomaly score is naturally the distance to c.

$$s(x) = ||\phi_\theta(x) - c||^2 \tag{7.4}$$

Here the aim is to compress all points to map onto a central point C and those that cannot be are deemed outliers.

Deep Clustering for Anomaly Detection Deep Embedded Clustering (DEC) [180] is one of the earlier deep clustering methods that combines representation learning with clustering using a clever self-supervision approach. Recently this work was extended to perform outlier detection [181].

The distance a point is from its closest centroid $\{c_1, ...c_K\}$ is naturally an anomaly score $s(x)$:

$$s(x) = \frac{\min_{k \in [1,K]} ||\phi_{\theta_e}(x) - c_k||^2}{\max_{j \in [1,n] \wedge m_j = k} ||\phi_{\theta_e}(x_j) - c_k||^2} \tag{7.5}$$

where $m_j = k$ denotes instance x_j belongs to cluster c_k, K denotes the total number of clusters, and $\phi_{\theta_e}(x_i)$ is the deep learner embedding function.

The core idea here is an extension to the one-class AD method mentioned earlier but extended to k clusters.

7.3.2. Causes Beyond Incompressibility. The above states that outliers are inherently points that the deep learner cannot compress. Hence it is natural to consider reasons why a deep learner cannot compress a group as being a key issue for unfairness. Here we conjecture three main reasons with the view they are related to biased outliers as shown in Figure 7.2.

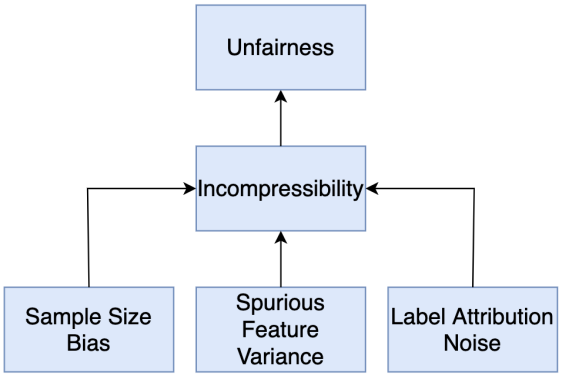


FIGURE 7.2. A Diagrammatic view of the expected reasons behind biased outlier detection.

Group Underrepresentation. Here we have a group that is relatively rare in the dataset but has some unique properties so the deep learner cannot compress it well. For example in Figure 7.1 many outliers are black but they only consist of under 15% of the dataset hence the deep learner uses the limited encoding space to encode more populous features.

Spurious Features for Groups. In this situation, the group has a property that is not critical for the outlier detection task but is highly variable. For example in Figure 7.1 many groups who are over-represented in the outliers wear different styles of hat.

Group Labeling Noise. Here the labeling of the group is inaccurate and hence can be a reason the group is labeled as being overly abundant in the outlier group. For example in Figure 7.1 the second to the bottom line of outliers all have the tag `Male` but this is erroneous.

7.3.3. Measurements of Unfairness and Four Properties. Before discussing our empirical results, we first define how each of the properties and how anomaly unfairness is measured. Many of these metrics are the maximum between some expression and their reciprocal. This is because the presence of a tag is equally important as the absence of a tag: for example, disparate

treatment of young people and disparate treatment of old (i.e. not young) people are equally important phenomena to study. We first describe how we measure unfairness for anomalies and then how we measure our four properties.

Anomaly DIR: The unfairness of an AD algorithm's output for particular group a is measured by the disparate impact ratio (DIR), which is defined [182]:

$$DIR(X, AD, a) = \max\left(\frac{P(AD(X) = 1|A = a)}{P(AD(X) = 1|A = \neg a)}, \frac{P(AD(X) = 1|A = \neg a)}{P(AD(X) = 1|A = a)}\right) \tag{7.6}$$

Here X is the dataset the AD algorithm (AD) has made predictions (normal vs anomaly) with $AD(x) = 1$ implying x is an anomaly, and a is the group in question. This is a natural choice for anomaly detection as it compares the rate at which different attributes are being flagged as anomalies, normalized by how often the rest of the data is considered anomalous. It is also the most widely used metric in fair unsupervised learning [183]. The range for this metric is $[1, \infty)$ with the larger the number the more unfairly group a is treated.

Incompressibility: To measure this feature, we extend the typical measure of reconstruction error into the novel metric of reconstruction ratio, which is defined:

$$RR(X, f, a) = \max\left(\frac{Loss_{MSE}(X, f(X)|A = a)}{Loss_{MSE}(X, f(X)|A = \neg a)}, \frac{Loss_{MSE}(X, f(X)|A = \neg a)}{Loss_{MSE}(X, f(X)|A = a)}\right) \tag{7.7}$$

Here X and a are the data used for AD and group again, with f being the autoencoder model (both encoder and decoder). The range of Equation 7.7 is therefore also $[1, \infty)$, where a higher number indicates that a group (or lack of a group) is harder to compress than the rest of the data. For example, a RR of 2 indicates that the attribute/group (or absence of the attribute/group) is twice as difficult to compress than the rest of the data.

Sample Size Bias (SSB): SSB (sometimes referred to as representation bias) is determined by the proportion of that tag or lack in the dataset X and is measured as [184]:

$$SSB(X, a) = max(P(A = a|X), P(A = \neg a|X)) \tag{7.8}$$

Where X and a are again the data and the group in question. Because all groups are binary (or encoded as one-hot encoding), the range of this metric is $[0.5, 1]$, with 0.5 indicating perfect balance of the group (i.e. males and females are equally likely) and 1 indicating that the group is always on or always off. Clearly, most groups will fall between these two extremes.

Spurious Feature Variance (SFV): SFV refers to the amount of variance in the background objects in the image and is measured as a proportion of the reconstruction error of the image:

$$SFV(X, f, a, b) = 1 - \max\left(\frac{Loss_{MSE}(X[b], f(X)[b]|A = a)}{Loss_{MSE}(X, f(X)|A = a)}, \frac{Loss_{MSE}(X[b], f(X)[b]|A = \neg a)}{Loss_{MSE}(X, f(X)|A = \neg a)}\right) \tag{7.9}$$

Where X is the data, f is the autoencoder, a the tag, and b is a bounding rectangle around the foreground/focus of the image (i.e. the face), either provided by the data or estimated [185]. As the denominator is clearly always greater than or equal to the numerator, SFV ranges between $[0, 1]$, where higher values indicate that more error comes from spurious features.

Attribute Label Noise (ALN): This is a metric of how noisy the labeling of a particular group is, as provided by the academic literature([186] for CelebA and [185] for LFW). Some groups such as Gender tend to have very low ALN, whereas other tags have very high ALN such as Blurry [185]. We define ALN as:

$$ALN(X, a, a^*) = 1 - (P(a = a^*|X) + P(\neg a = \neg a^*|X)) \tag{7.10}$$

Where X is the data, a the group in question, and a^* the true label for the group. This property has a range $[0, 1]$ where the higher the value the less reliable the group labeling.

7.4. Experimental Results - Who Is AD Unfair To?

Here we answer the question: Who are the groups of individuals most adversely affected. Following this, we explore more nuanced inquiries, such as whether the unfairness is attributable solely to the data, the algorithm, or a combination of both. In the subsequent section, we aim to investigate the underlying reasons for the unfairness inherent in AD.

Our experiments consist of two core AD algorithms: A reconstruction-based autoencoder anomaly detection algorithm (hereby referred to as AE) and Deep one-class SVDD. As mentioned earlier, clustering-based AD is a generalization of one-class algorithms and the AE methods. Our datasets consist of the CelebFaces Attributes Dataset [187] (the 50,000 instance version to reduce compute) which consists primarily of popular individuals in the movies, music, or arts whilst our Labeled Faces in the Wild [139] consists of approximately 13,000 instances and includes a wider variety types of popular individuals such as politicians, sports stars, and criminals. Attribution is provided by [185]. These two datasets were chosen as they are well-annotated, including analyses of labeling error, and have been extensively studied. Among all of our datasets, we test a total of 63,233 facial images covering 111 attribute tags. We examine each algorithm individually for a total of 222 data points.

For each dataset and algorithm, we determine the unfairness of each group using the Anomaly DIR. Results are collected over five random-initializations of the network and the median results for each property are reported. The list of all raw results is in the Appendix, but here we outline some key insights.

The Algorithms are Overwhelming Fair to Most Groups. In total amongst both the two algorithms and two datasets there are 222 groups and a frequency distribution shows that overwhelming the algorithms are fair with respect to over 70% of the groups as shown in Figure 7.3. A score of less than 1.2 indicates that the ratio of the group in the anomalies is not more than 20% more than the rate of the other groups occurring in the anomalies.

However, there are significant examples of unfairness whose properties we now discuss.

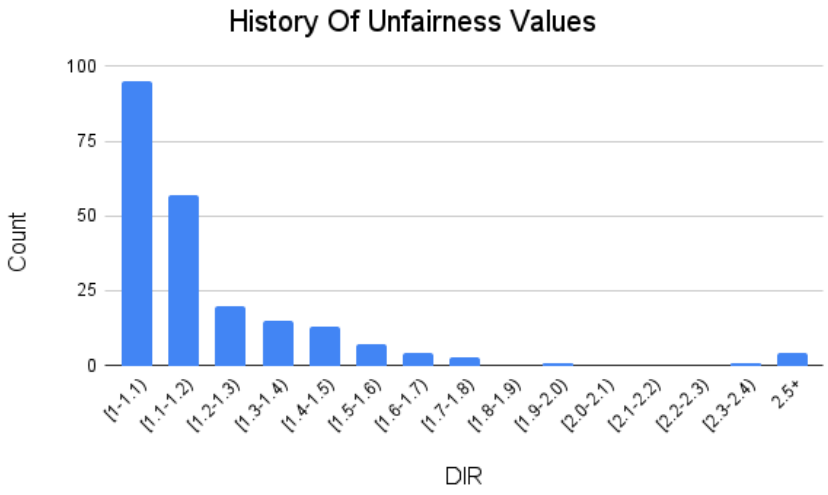


FIGURE 7.3. A frequency distribution of the Anomaly DIR score versus how often it occurs across all algorithms and datasets.

Few Groups Are Always Treated Unfairly. We found that there are several groups that are always (regardless of algorithm or dataset) treated unfairly but they are relatively rare. These include the groups centered around weight having the annotations `Chubby, Double-Chin` and those centered around very unusual image properties such as `Wearing-Hats`. This is not unexpected given a very rare group with unusual properties (not shared by other groups) are unlikely to be well compressed. In total less than 2% of all groups are treated unfairly all the time.

Unfairness Varies Due to Both Algorithm and Dataset. A more likely occurrence is that some groups are treated very unfairly but only for some datasets and some algorithms. Table 7.1 shows in bold groups treated unfairly (the Anomaly DIR is shown in parentheses) but only for that dataset and algorithm combination. For other algorithm-dataset combinations, they are treated fairly as the Table shows. This result is surprising and shows the strong interaction between the algorithm and the data. Consider that the AE labeled the the ¬`No Beard` (reported as "`Beard`") in the CelebA dataset at a rate over 3 times greater than the other groups. Yet, the SVDD algorithm on the very same dataset produced just a 1.27 DIR for the `Beard` group, and in the LFW dataset both algorithms the DIR was below 1.2.

The More Focused The Dataset The More Likely Unfairness Can Occur. When we aggregated all fairness DIR scores (see Appendix) for each group and all algorithms we found that

	CelebA	LFW
AE	**Beard (3.244)**	Beard (1.061)
	Senior (N/A)	**Senior (1.8)**
	Gray Hair (1.053)	Gray Hair (1.028)
	Unattractive (1.075)	Unattractive Man (1.158)
SVDD	Beard (1.267)	Beard (1.0876)
	Senior (N/A)	Senior (1.0018)
	Gray Hair (2.449)	Gray Hair (1.197)
	Unattractive (1.094)	**Unattractive Man (1.566)**

TABLE 7.1. Examples of groups treated unfairly only for a particular algorithm and dataset interaction. The Fairness DIR is reported in parentheses and indicates the relative over-abundance of the group in the anomalies. The tag being treated unfairly in these cases is in bold. For example, people with a Beard are 3.224 times more likely to be an anomaly than a normal instance for the AE algorithm applied to the CelebA dataset, though people with beards are treated relatively fairly otherwise. Not that "Senior" is not a tag in CelebA and is therefore absent from the in these cells.

the CelebA dataset (DIR = 1.4) causes significantly more unfairness than the LFW dataset (DIR = 1.13).

This is likely due to the CelebA dataset having a much more focused selection bias as it is limited to people who are overwhelmingly in the arts (film, television, music) whereas the LFW dataset consists of a larger representation of popular people. Hence, the definition of normality learned is very specific and there are many ways to deviate from the norm.

Examples of groups that are found to be unfairly treated in the CelebA dataset but NOT the LFW dataset are: `Wearing Hat, Big Nose, Eye-Glasses, Goatee, Wavy-Hair`.

The More Focused The Algorithm The More Likely Unfairness Can Occur.

Similarly, the way the algorithm defines normality is influential in who it identifies as an anomaly. The SVDD algorithm has the strictest definition of normality as it attempts to find just one group of normal instances (centered around c see equation 7.3) whereas the AE algorithm with k encoding nodes can in practice (assuming perfect disentanglement) find at least k definitions of normality. Hence not surprisingly the SVDD algorithm is more unfairer than the AE algorithm as shown by the histogram of unfairness for both algorithms in Figure 7.4.

7.5. Experimental Results - Why is AD Unfair

Here we attempt to experimentally answer the following questions:

- How strong are our four properties correlated to unfairness?
- How are our four properties related to each other and in particular is there a hierarchical structure to them?
- How can these properties be combined to create a model to explain unfairness in anomaly detection?

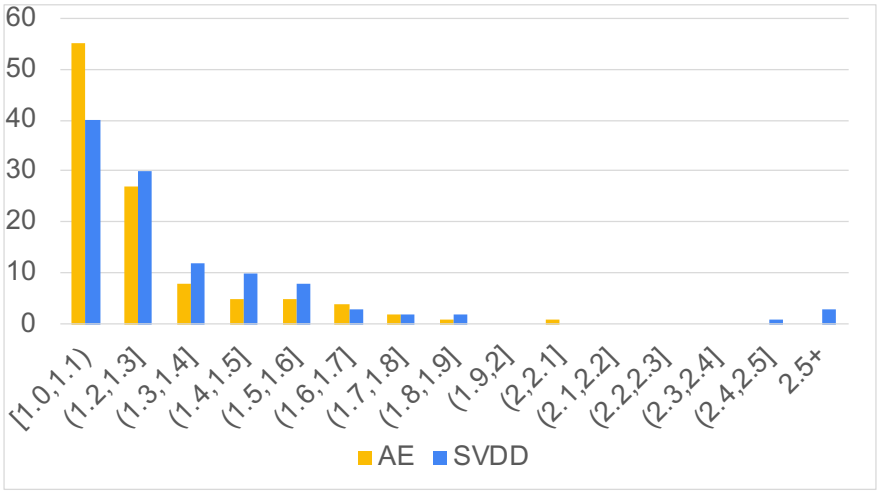
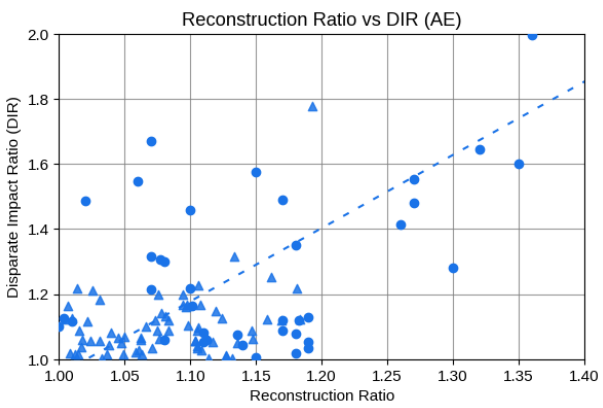
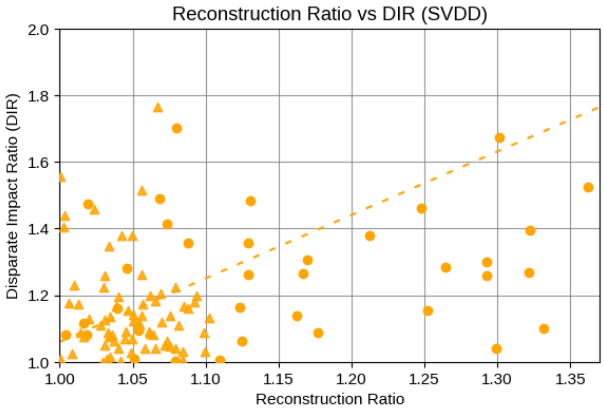


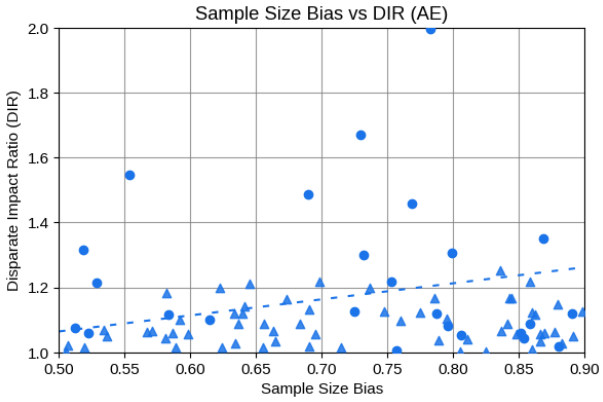
FIGURE 7.4. A frequency distribution of the Anomaly DIR score by algorithm. We see that the AE with a more flexible definition of normality is more fair.



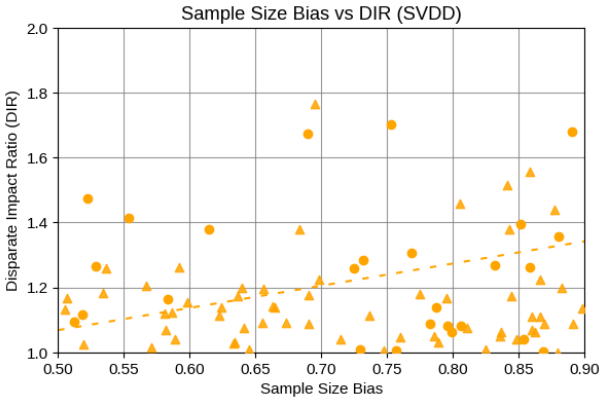
(a) Corr: 0.568, RSQ: 0.334



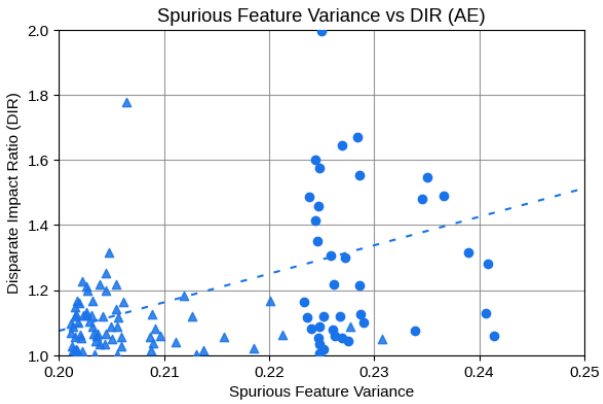
(b) Corr: 0.523, RSQ:0.273



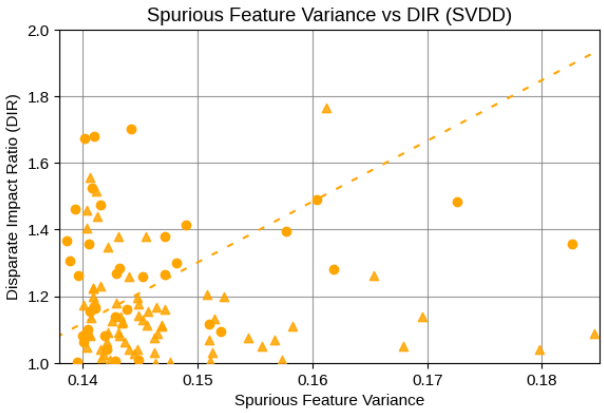
(c) Corr: 0.220, RSQ:0.114



(d) Corr: 0.251, RSQ:0.128



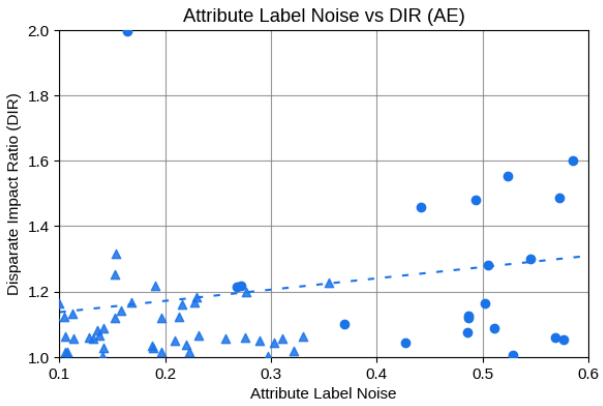
(e) Corr: 0.337, RSQ:0.148



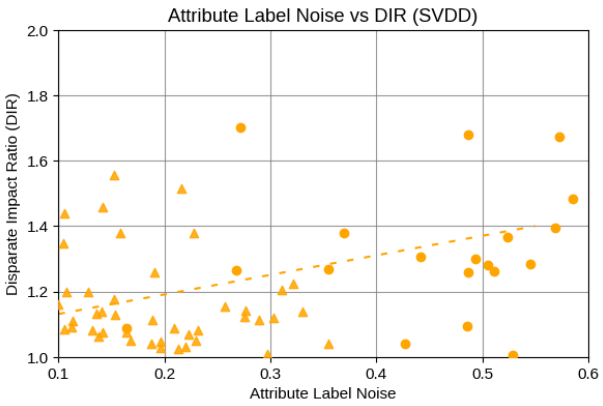
(f) Corr: 0.473, RSQ:0.224

FIGURE 7.5. (Figure continues on next page)

.



(g) Corr: 0.261, RSQ:0.167



(h) Corr: 0.328, RSQ:0.108

FIGURE 7.5. Plot of different properties against their DIR (unfairness) with the larger the value the more of the property/unfairness. Trendlines are created by minimizing R^2 values. Each mark represents one group. Color denotes algorithm (blue for the AE anomaly detector and orange for the SVDD anomaly detector) and mark denotes dataset (circle for CelebA, triangle for LFW).

7.5.1. Relationship between Unfairness and Each Property. Our experiments (see Figure 7.5) demonstrate strong (Pearson) correlations and moderate to strong RSQ (R-squared values of the regression trendline) for each of the properties studied. Each plot shows the results for two datasets (CelebA and LFW) with each data point representing a group of individuals. A positive trend line indicates positive Pearson correlation (see sub-titles of plots for exact values) and we see that incompressability is the most strongest property correlated with unfairness, then Spurious features, then Attribute label noise, and finally Sample Size Bias. This is an interesting result as earlier seminal results showed that AD using facial images [167] was unfair due to an under-representation of black people and males in the underlying datasets.

However, it is also clear that no individual property explains unfairness completely by itself. This is shown as each graph has points that not only do not fit the trendline, but are contradictory to the relationship implied by the overall data. Further investigation (see next subsection) reveals that when one property fails to explain why that attribute is anomalous, another one typically will.

For example, the group `Bags Under Eyes` (from CelebA) under the AE model has a reconstruction ratio of only 1.077 (it is easy to compress), but a DIR of 1.31 (it is treated unfairly). Following the trend, the expected reconstruction ratio at a group with this DIR would be approximately 1.17. Further, this group has only 20.1% representation, though looking at the DIR one would expect

only half that. This group, however, is explained by the spurious feature variance, as it sits nearly perfectly on the trendline. Similarly, the group `Gray Hair` (from LFW) under Deep SVDD was towards the far end of spurious feature variance at 0.180, but has extremely low anomaly DIR score at 1.04 (i.e. was treated fairly), though it sits just above the trendline for attribute label noise at 1.05.

A full list of these attributes and their squared error for all trendlines is available in the Appendix, and one can see that every tag can be explained by at least one of these properties with high fidelity, with the average sum of square errors being only 0.00351 (std 0.006498), supporting our claim that unfairness in anomaly detection setting can be explained by one of these four properties. This claim is rigorously tested in Section 7.5.2.1.

7.5.2. Relationship between Multiple Properties. We also examine the correlation between the different properties. This analysis is useful in examining potential redundancies and creating our model of unfairness for anomaly detection. Figure 7.6 examines these relationships. Some features are, indeed, positively correlated with each other, though none have high enough correlation to suggest that they are redundant with each other. In the subsequent subsection, we examine this claim more rigorously via a hypothesis test.

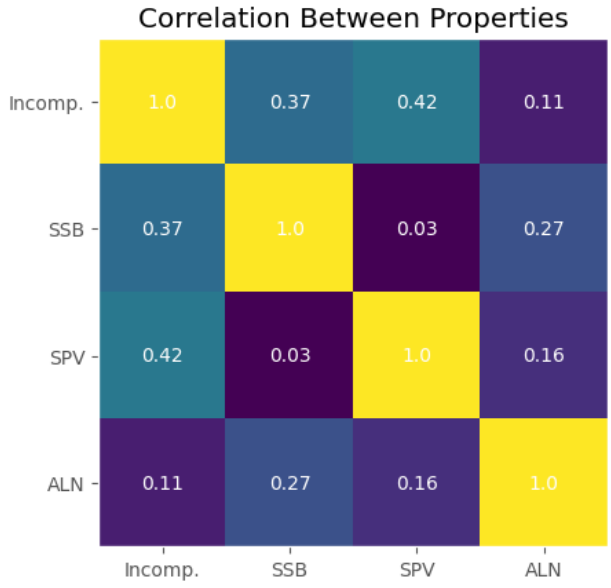


FIGURE 7.6. Correlation matrix for all four properties of the model. Pearson correlation is written in each box and is consistent with color (yellow is large, purple is small).

7.5.2.1. *Hypothesis Testing of Relationship Claims.* In order to test our claims, we create four hypotheses that we verify through hypothesis significance-testing. Those are:

- H1: No individual property is sufficient to always explain unfairness.
- H2: The properties, when combined into a multiple regression, are sufficient to explain unfairness.
- H3: No properties of the multiple regression are redundant and all are needed.
- H4: The results of H2 are significant in that when one property fails to predict unfairness, another does.

Null hypothesised $H1_0 - H4_0$ are constructed straightforwardly. To create the significance test for H1, we perform an F-test on individual regression models crafted from the relationship between each property and DIR. The results of this F-Test (visualized in Figure 7.7) indicate that individual properties are reasonable though comparably weak predictors of unfairness, with P-values ranging from 0.0137-0.0986 for the AE model and 0.0279-0.0571 for Deep SVDD. Therefore, we reject the null hypothesis $H1_0$ and validate hypothesis H1.

To test hypotheses H2 and H3, we construct a multiple-regression model. Specifically, this is a stacked multiple regression where the meta-function selects the best individual model for the datum. To validate H2, we create such a multiple-regression using all four of the properties (the "full" model). This yields P-Values of 0.00589 for the AE model and 0.0127 for Deep SVDD, significantly lower than those of the respective single-regression models, and indicating that using all four properties is sufficient to explain how unfairness occurs. We reject the null hypothesis $H2_0$ and validate hypothesis H2.

For H3, we conduct a similar experiment except we leave one property out. In every case, the resulting multiple regression models were worse than the full model, with P-Values ranging from 0.00674-0.0109 for the AE model and 0.0138-0.0164 for Deep SVDD, all greater than that of the full model, indicating that every property is necessary and none are redundant. We reject the null hypothesis $H3_0$ and validate hypothesis H3.

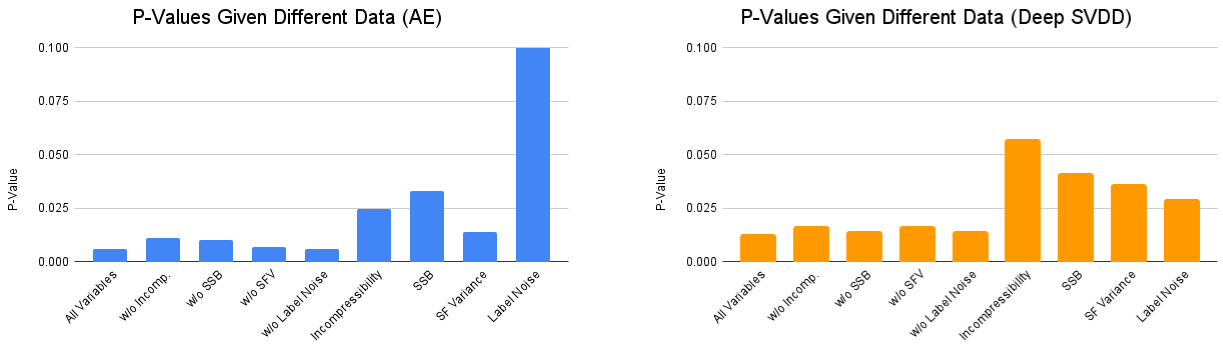


FIGURE 7.7. P-Values for the hypotheses H1-H3. The leftmost bar demonstrates that, when all properties are considered, unfairness can be predicted with a very high degree of precision, rejecting the null hypothesis $H2_0$. The next three rows demonstrate that the model is not as powerful if one property was left out, rejecting the null hypothesis $H3_0$. Finally, the higher P-values for the simple regressors indicate that no single feature can be used as a model of unfairness, rejecting null hypothesis $H1_0$.

One may object to the multiple-regression models used above, given that the model as described will monotonically increase in predictive power given more properties. It is important to note that this model matches the central claim of this paper - that unfairness with respect to a group occurs because of one of the four properties described, though one may still be wary of the statistical significance of the reported results given the technique. To resolve these concerns, we demonstrate that our model is not just combining the predictive power of four different already powerful predictors, but rather when one model fails it is because it is explained by one of the other properties.

To validate this claim, we construct fabricated distributions similar to those of Figure 7.5. Specifically, unfairness is kept the same, and we create distributions of random fake data which has the same correlation and RSQ as all of those shown. This is accomplished by, for each property, finding random points (sampled across a uniform distribution) along the X-axis, giving them fabricated values perfectly in line with the correlation, and then adding noise such that the correlation is maintained and the RSQ matches that of the actual measured properties. Then, we create the same full model of the multiple regression and measure the P-value. We repeat this process 10,000 times to get 10,000 such distributions.

The distributions therefore should be statistically similar to our real data, but there is no reason to believe that when one of the fabricated models fails, another will explain the unfairness. To validate hypothesis H4, we measure the number of times the fake distributions produce P-values under that of the real data. If the statically similar fabricated data cannot match the predictive performance of our models, this would validate hypothesis H4.

In the case of the AE model, the fabricated data averaged a P-value of 0.0194 with a standard deviation of 0.00629 and never beat the full model's P-value of 0.00589. Similarly, the model simulating Deep SVDD's data yielded an average P-value of 0.0173 with a standard deviation of 0.00304. Out of the 10,000 trials, only 5 yielded lower P-values. Therefore, we reject the null hypothesis $H4_0$ and validate hypothesis H4. Our model does not simply take four independent good predictors of anomaly and get good statistical results but rather holds the property that when one fails, another property explains it.

7.5.3. A Proposed Model Of Unsupervised Unfairness Relationships. Given the resulting hypothesis tests, we craft our model of unfairness in unsupervised learning. Figure 7.8 provides a graphical representation of this model. Edges between properties indicate a relationship (binarized to be correlated at ≥ 0.15). This is supported by the high correlation between each of these properties and unfairness (Figure 7.5), the result that the properties together form a uniquely powerful multiple-regression to explain unfairness (H2, H4), that no single feature could do this alone (H1), and that no property is redundant (H3).

7.6. Discussion and Conclusion

We study the intersection of the controversial deep AD algorithm with facial imaging data to address the "Who" and "Why" questions. We found that overwhelmingly both auto-encoder and one-class deep AD algorithms are fair to most groups. However, due to the compression-based focus, they are unfair to some sub-groups.

With regard to the "Who" question we found that it was rare to be consistently unfair to the one group and instead unfairness was due to the interaction of the data and the algorithm. In particular, the more focused the dataset and algorithm the more unfairness was found.

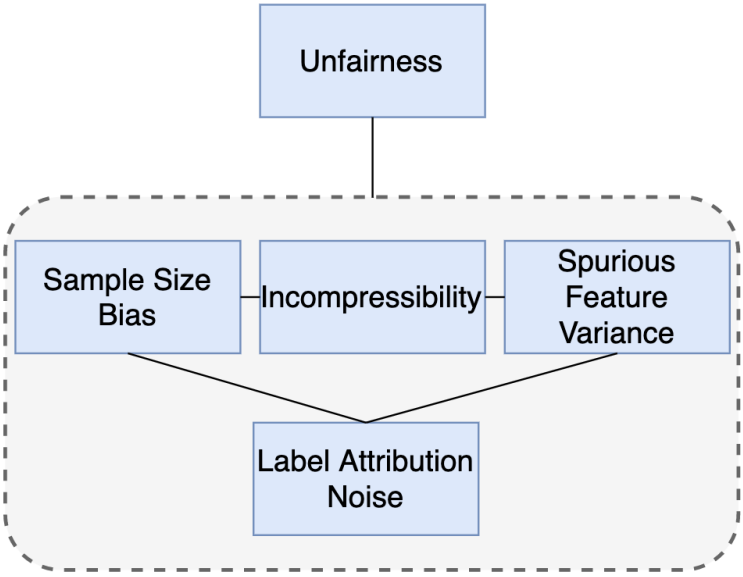


FIGURE 7.8. Our model of unfairness determined from our stacked multi-regression model. Compare with the expected model without any analysis in Figure 7.2.

Our study of the "Why" question aimed at developing a deeper understanding on the effect of data related factors on the fairness as well as detection performance of OD algorithms. We postulated four hypotheses and found all to be statistically significant by rejecting the null hypothesis. The first hypothesis is that no single property alone is sufficient to explain unfairness. The second hypothesis is when combined the properties can explain unfairness. The third hypothesis is that all properties are relevant and none are redundant and finally, the fourth hypothesis is that the combination of properties is meaningful beyond the predictive power of each individual property.

Limitations. The use of groups may have varying degrees of applicability to real-world fairness scenarios. For example, some groups such as `Male, Black` and `Young` correspond to legally recognized protected classes [188, 189], while others such as `Goatee, Wearing Hat` and `attractive` may not. However, we believe that this study still provides meaningful insights into the mechanism of unfairness with respect to different people. Real-world protected attributes may be of varying degrees of visibility, as do our groups, and our analysis reflects this.

Future work. Remediation strategies to improve fairness are left out of scope of our investigation. We briefly discuss them here. Fairness interventions are typically grouped into three: pre-, post-, and in-processing strategies, which respectively, modify the input data, modify the output scores or decisions, and account for fairness during model training.

As we showed, AD unfairness can stem from algorithmic bias alone in the face of natural heterogeneities in the data among or within groups. When this is the case, pre-processing strategies become voided as it is not clear how to modify organic, unbiased data. Post-processing could select different thresholds for each group separately, as in [190, 191], where the group-specific thresholds could either be "natural" cut-off values, or selected to optimize demographic parity if it is a desired fairness metric. Note that metrics that involve true labels cannot be optimized due to lack of any ground truth during training. In-processing techniques are also limited to only enforcing demographic parity, which as we showed, remains susceptible to unfairness. One such strategy that has not been applied to OD is decoupling, as in [192, 193], where a different detector is trained for each group, while optimizing a joint loss.

We remark that post-processing and decoupling exhibit treatment disparity as they both assume it to be ethical and legal to use the sensitive attribute at test (decision) time - in particular, to select which threshold or detector to employ on a given new sample. When there are differences *among* groups, coming to terms with treatment disparity might be the only get-around to mitigating disparate impact, as argued previously [194]. These solutions, however, do not address unfairness against heterogeneous subpopulations *within* groups, i.e. within-group discrimination. Here, one direction is to explore clustering-based OD algorithms. Alternatively, establishing a more nuanced or granular sensitive attribute, labeling each subpopulation differently.

CHAPTER 8

Beyond Data Bias: Proof of Algorithmic Fairness Challenges in Neural Networks

Abstract As machine learning and data science are used in increasingly high-stakes domains such as criminal justice, healthcare, and finance, it is pertinent that machines are not only accurate but also equitable. In previous years, many hoped that machines would ameliorate issues of prejudice in such critical decisions as machines are blind to identity and only see the relevant data fed to them. Unfortunately, this is not true, as researchers note that machines can produce bias against certain groups just as people do. This has led researchers to believe that while the algorithms and machines themselves may not hold an inherent bias against a group, data collected by humans may share the biases of those humans. While bias in data certainly exists, we believe that this is only one part of the picture, and reject the idea that decisions about the algorithm such as model architecture, hyperparameters, and optimizers cannot have bias in and of themselves. In this paper, we provide a rigorous mathematical proof that demonstrates that even when all other elements such as data, protected status breakdown, and target demographics are equal, bias still exists and is inherent to these algorithms.

8.1. Proof

THEOREM 8.1.1. *A regularized artificial neural network with a bounded loss function has bias inherent to its architecture, optimizer, and/or hyperparameters, assuming that it will be allowed to train to convergence.*

COROLLARY 8.1.1. *The bias inherent to a network is predictable and calculable.*

PROOF.

LEMMA 8.1.1.1. *The parameter space of a regularized neural network is finite and bounded.*

PROOF. Consider the loss function of a regularized neural network, given by:

$$Loss = Loss_{error} + Loss_{regularization} \tag{8.1}$$

A parameter w is learned through the negative gradient of this loss function:

$$w = w - \alpha \nabla Loss \tag{8.2}$$

Where α is the learning rate. From Equation 8.2, it is clear that a parameter will increase in magnitude whenever the gradient of the cost function is negative. In the case of the gradient of $Loss_{error}$ being negative, the parameter w will increase in value if and only if the gradient of the regularization term is also negative. That is, the parameter w can only grow in magnitude if $Loss_{error} > Loss_{regularization}$. Typically (and stated as an explicit assumption of Theorem 8.1.1), the error component of the loss function is bounded within some range. For example, for a binary classification problem where output space is either 0 or 1, the greatest error possible for a single instance is when the output of the network is 0 and the label is 1, or when the output of the network is 1 and the label is 0, in either case $Loss_{error}$ is bounded between $[0, 1]$. Without loss of generality, let the maximum value of the gradient of the loss due to error be expressed as E_{max} and the minimum (most negative) by expressed as E_{min}. Since the gradient of $Loss_{error}$ is bounded by $[E_{max}, E_{min}]$ and the regularization term $Loss_{regularization}$ is unbounded for some arbitrarily large parameter w, the parameter w will necessarily decrease in magnitude if:

$$Loss_{regularization} \geq E_{max} \tag{8.3}$$

Or:

$$Loss_{regularization} \leq E_{min} \tag{8.4}$$

For example, in the case of L2-regularization given by:

$$Loss_{L2} = \lambda w^2 \tag{8.5}$$

The magnitude of the parameter w will necessarily decrease if:

$$w \geq \sqrt{\frac{E_{max}}{\lambda}} \tag{8.6}$$

Similarly, the lower bound can be calculated by inverting the inequality and replacing E_{max} with E_{min}, meaning that every parameter w is bounded between $[\sqrt{\frac{E_{min}}{\lambda}}, \sqrt{\frac{E_{max}}{\lambda}}]$. Let the minimum and maximum value for every parameter be labeled w_{min} and w_{max} respectively. Since one of the assumptions of Theorem 8.1.1 is that the neural network is trained to convergence, every parameter of the network will be between these bounds.

QED Lemma 8.1.1.1. □

LEMMA 8.1.1.2. *Given a particular architecture, the mappings between input and output space are bounded.*

PROOF. Given a set of parameters θ, a neural network deterministically maps an input space X to an output space Y by:

$$f(X, \theta) \rightarrow Y \tag{8.7}$$

By Lemma 8.1.1.1, the possible values for each parameter $w \in \theta$ are finite and bounded, ergo all mappings between input space and output space for a particular architecture of a neural network are bounded.

QED Lemma 8.1.1.2. □

LEMMA 8.1.1.3. *Neural networks will converge to a particular local minima by some probability.*

PROOF. Let the function $Loss(f, X, \theta)$ be defined as the loss for some data X given the network f with parameters θ.

A local minima in parameter space is defined as a configuration of parameters θ_{min_i} such that:

$$\begin{gathered} \forall \theta_n Loss(f, X, \theta_{min_i}) \leq Loss(f, X, \theta_n) \\ s.t. ||\theta_n - \theta_{min_i}|| < \epsilon \end{gathered} \tag{8.8}$$

For all arbitrarily small ϵ where θ_n is some other set of parameters.

As proven in Lemma 8.1.1.2, mappings between input and output space is bounded for a particular architecture. Therefore, for every mapping $f(X, \theta_i) \rightarrow Y$ there exists a loss given by $Loss(f, X, \theta)$. Since the parameter space is finite, there exists either:

- A finite number of local minima $\Theta_{min} = \{\theta_{min_0}, \theta_{min_1}, ...\theta_{min_k}\}$

- A finite number of local minima regions, that is, a set of bounds $\{w_{min_0}, w_{max_0}, ..., w_{min_x}, w_{max_x}\}$ for all x parameters of the network's architecture for which $\forall \theta_{min_i} s.t. \forall w_n \in \theta_{min_i} w_n \in [w_{min_n}, w_{max_n}], \theta_{min_i}$ is a local minimum by Equation 8.8.
- Both of the above two conditions. A visual for both these types of minima is provided in Figure 8.1.

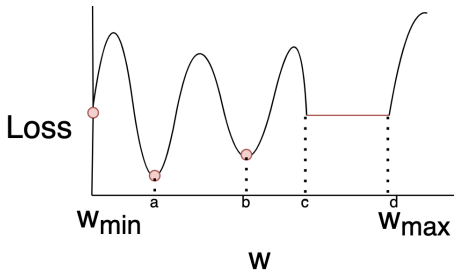


FIGURE 8.1. Example of the two types of minima with respect to a single parameter, w. Here, minima occur when $w \in \{w_{min}, a, b\}$ (individual local minima) or when $w \in [c, d]$ (local minima region).

For simplicity, we will use the notation of a set of local minima to represent the set of all local minima and local minima regions: $\Theta_{min} = \{\theta_{min_0}, \theta_{min_1}, ...\theta_{min_k}\}$.

Given an initial parameterization and a deterministic optimizer, a neural network will converge to some $\theta_i \in \Theta_{min}$ with either probability of 1 or 0. Given an initial parameterization and a stochastic optimizer, the network will converge to some $\theta_i \in \Theta min$ with some probability between $[0, 1]$ based on the particular optimizer and loss landscape. In either case, the network will converge to any one of the local minima in the set Θ_{min}.

QED Lemma 8.1.1.3. □

LEMMA 8.1.1.4. *The fairness of a neural network is finite and bounded.*

PROOF. For some fixed population D with (a) protected status variable(s) PSV, fairness for a particular algorithm and collection of parameters θ can be given by:

$$Fair(f, D, PSV, \theta) \tag{8.9}$$

where the $Fair()$ function is whichever fairness metric one chooses (e.g. disparate impact, equalized odds, predictive equity, etc). By Lemma 8.1.1.3, the trained neural network has converged

to some $\theta_{min_i} \in \Theta_{min}$. $\forall \theta_{min_i} \in \Theta_{min}, \exists Fair(f, D, PSV, \theta_{min_i})$, $\therefore$ the fairness of a network is bounded between $[argmin_{\theta_{min_i}} Fair(f, D, PSV, \theta_{min_i}), argmax_{\theta_{min_i}} Fair(f, D, PSV, \theta_{min_i})]$

QED Lemma 8.1.1.4 □

By Lemma 8.1.1.4, there are a finite number of values for fairness that a network can achieve based on which local minima it converges to. By Lemma 8.1.1.3, every local minima can be converged to with some probability. Therefore, one can calculate the expected value for fairness of a network in the following way:

$$\begin{aligned} &ExpectedFairness(f, D, PSV) = \\ &\sum_{i=1}^{n} Fair(f, D, PSV, \theta_{min_i}) * \\ &P(\theta_f = \theta_{min_i}) \end{aligned} \tag{8.10}$$

Where $P(\theta_f = \theta_{min_i})$ is the probability that the model f of converging on the local minima θ_i. This can be calculated by integrating across every dimension of parameter space for w_{min} to w_{max} (or the maximum and minimum values for a random initialization). That is, the expected fairness of a particular neural network (before it is trained) is given by:

$$\begin{aligned} &ExpectedFairness(f, D, PSV) = \\ &\sum_{i=1}^{n} Fair(f, D, PSV, \theta_{min_i}) * \\ &\int_{w_0=w_{min}}^{w_k=w_{max}} \int_{w_k=w_{min}}^{w_1=w_{max}} \dots \int_{w_k=w_{min}}^{w_k=w_{max}} \\ &P(\theta_f = \theta_{min_i} | \theta_{init} = \{w_0, w1, ...w_k\}) * \\ &P(\theta_{init}) \\ &dw_0 dw_1 \dots dw_n \end{aligned} \tag{8.11}$$

Equation 8.11 demonstrates the predictability and calculability aspect of Corrolary 8.1.1.

Therefore, for two networks f_0 and f_1 of different architecture, optimizers, or hyperparameters $\neg\Box ExpectedFairness(f_0, D, PSV) = ExpectedFairness(f_1, D, PSV)$ (read as: "the expected

fairness of f_0 is not necessarily equal to that of f_1"). In other words, all other variables being equal, some of the bias of a learner is due to the specifics of the algorithm itself.

QED Theorem 8.1.1 & Corollary 8.1.1.

□

8.2. Approximation To Expected Fairness

As demonstrated in Theorem 8.1.1, Equation 8.11 calculates the expected fairness for a particular algorithm. Unfortunately, this equation is intractable with respect to the number of parameters of the model, which tends to be quite high. In this section, we will discuss an approximation to this equation that is far less computationally expensive. First, we consider the approximations from two different perspectives: the search for the set Θ_{min} of all local minima for the algorithm f (and, as a consequence the fairness for each local minima) and the integration over parameter space for finding the probability that one will end up in a particular local minima.

To simplify this process, we will break up Equation 8.11 into three parts, labeled a b, and c:

$$a(\theta_{min_i}) = Fair(f, D, PSV, \theta_{min_i}) \tag{8.12}$$

$$b(\theta_{init}) = P(\theta_{init}) \tag{8.13}$$

$$c(\theta_{min_i}, \theta_{init}) = P(\theta_f = \theta_{min_i} | \theta_{init}) \tag{8.14}$$

In the proceeding subsections, we will discuss how to approximate each of these and then combine the two into an approximation algorithm.

8.2.1. Approximating The Number Of Local Minima & Fairness Landscapes. Equation 8.11 assumes that we perform an exhaustive search or mathematical optimization over parameter space to find all local minima. Instead, we propose approximating this search by constructing a two-dimensional loss landscape for the network as presented in [195], then finding all local minima with the simplified space. After constructing the loss landscape, we search for local minima by

determining the set Θ_{min} in the landscape satisfy Equation 8.8 within some small ϵ. This also simplifies the issue of minimal regions because it approximates those regions as a series of neighboring minima.

From there, we propose creating a novel visualization: the fairness landscape. Every point in the two-dimensional loss landscape represents a particular set of parameters θ_i. Using the conventions outlined in Equation 8.9, one can find the fairness for every point in the loss landscape and construct a visualization of how fairness changes for different sets of parameters. Then, we can construct the set

$$Fairness_{min_i} = \{Fair(f, D, PSV, \theta_{min_i}\} \forall \theta_{min_i} \in \Theta_{min} \tag{8.15}$$

which is an approximation to the set of all fairness values that can be achieved by the network if allowed to train to convergence.

8.2.2. Approximating The Probability of Arriving At Local Minima. Equation 8.11 also requires a large multiple integral, requiring one integral per parameter in the network. Semantically, this integral finds the probability that the network ends up in a local minima given an initialization, multiplied by the probability of that initialization. Instead, we propose approximating this integral by performing a hill-climbing search over the loss landscape found in Section 8.2.1. Given the two-dimensional space, we sample initialization θ_{init} and perform hill-climbing optimization until the optimization ends up at one of the local minima θ_{min_i}. If we are discussing a deterministic optimizer, we set the probability $P(\theta_f = \theta_{min_i} | \theta_{init})$ to 1, and $\forall \theta_{min_j} \in \Theta_{min} s.t. i \neq j P(\theta_f = \theta_{min_j} | \theta_{init})$ to 0. For a stochastic optimizer, we can set those values to the fraction of times the network converged to the minima over the total number of iterations for each minima in Θ_{min}.

The final part of the equation, calculating $P(\theta_{init})$ is calculated exactly, and is dependent on the particular initialization routine. Frequently, initializations are random weight vectors around 0 sampled uniformly [196]. In this case, one can calculate these values by taking k initializations spaced uniformly in the two-dimensional loss landscape, and the probability is simply $1/k$.

8.2.3. An Algorithm For Approximating Fairness Of An Algorithm. We replace the integral of Equation 8.11 with a summation, leading to the final approximation of:

$$\sum_{i=1}^{n} a(\theta_{min_i}) * \sum_{x} \sum_{y} b(\theta_{init=(x,y)}) c(\theta_{min_i}, \theta_{init} = (x, y)) \tag{8.16}$$

Where x and y are the dimensions of the parameter space in the loss landscape. From the preceding subsections, we construct an algorithm to approximate 8.11.

- Construct a two-dimensional loss landscape to approximate parameter space and find all local minima Θ_{min}
- Given the set of local minima, find the set $Fairness_{min}$ by Equation 8.15 to approximate Equation 8.12.
- Calculate Equation 8.13 by the specifics of the network's initialization procedure
- Sample k initializations and perform hill-climbing on the loss landscape to determine an approximation to Equation 8.14
- Use the above three components to calculate the approximation by Equation 8.16.

8.3. Empirical Evaluation Of Integral

To empirically validate the results of our proof, we demonstrate that solving the integral in Equation 8.11 can accurately predict the value of unfairness present in a network before training. Here, we solve the full integral, rather than using the loss landscape approximation. In this section, we discuss the runtime of the technique and demonstrate that our formula can accurately approximate the fairness of neural networks *before* training them, and use this to predict which of two neural networks will be more fair, demonstrating the correctness of the approach. The fairness metric used for this evaluation is statistical parity [182].

The use of sparse grid integration also allows us to create a novel visualization: the fairness-loss landscape. Similar to loss landscapes [195], the fairness-loss landscape presents the loss for the network given the values for different parameters along a grid, however we also overlay this with the value of fairness for each of those points.

8.3.1. Runtime Analysis. As Equation 8.11 is a multiple integrals that requires an additional integral for every parameter, the calculation is intractable, as the runtime for solving or

computationally approximating an integral is intractable, rising in complexity exponentially with respect to the number of dimensions [197]. As such, we validate our results over small neural networks and solve the integral using sparse grid integration [198]. Assuming the grid is broken by an interval i over an area where each dimension of that area is given by $\mathcal{L}$, the runtime of computing Equation 8.11 is $O(\frac{\mathcal{L}}{i}^d)$.

8.3.2. Datasets & Models. Because of the exponential complexity of solving the integral, we intentionally use very small networks and low-dimensional data. Specifically, we estimate the fairness for two networks, both trained on the Law School dataset [199], one with two parameters (one weight from each feature) f_1 and one with three (one weight from each parameter and a bias term) f_2. After the integration is performed, the expected value of fairness compared to the average result of fairness for training the model roughly to convergence 100 times.

8.3.3. Results. Recall that to solve Equation 8.11, one needs three functions: $Fair(f, D, PSV, \theta_{min_i})$, $P(\theta_f = \theta_{min_i} | \theta_{init} = \{w_0, w1, ...w_k\})$, and $P(\theta_{init} = \{w_0, w1, ...w_k\})$.

As we are using sparse-grid integration, we estimate the function $P(\theta_f = \theta_{min_i} | \theta_{init} = \{w_0, w1, ...w_k\})$ using the generated grid by taking the value of the loss at each position in the grid and using its neighbors to approximate which local minimum a point initialized to that position would fall into. Similarly, the $Fair$ function is determined by the fairness value of the nearest element on the sparse grid. Initialization is uniform over $(\frac{-1}{\sqrt{2k}}, \frac{-1}{\sqrt{2k}})$ [200], so the probability of initializing at a certain point is given by $1/P$ where P is the number of points on the grid within the initialization bounds and 0 otherwise.

Using these three functions, sparse grid integration is performed. The expected fairness for the model f_1 is 0.1134 and f_2 is 0.1348, compared to the actual values of fairness 0.1107 and 0.1385, respectively. These values are very close, only differing by about 2.5%. Table 8.1 summarizes these results.

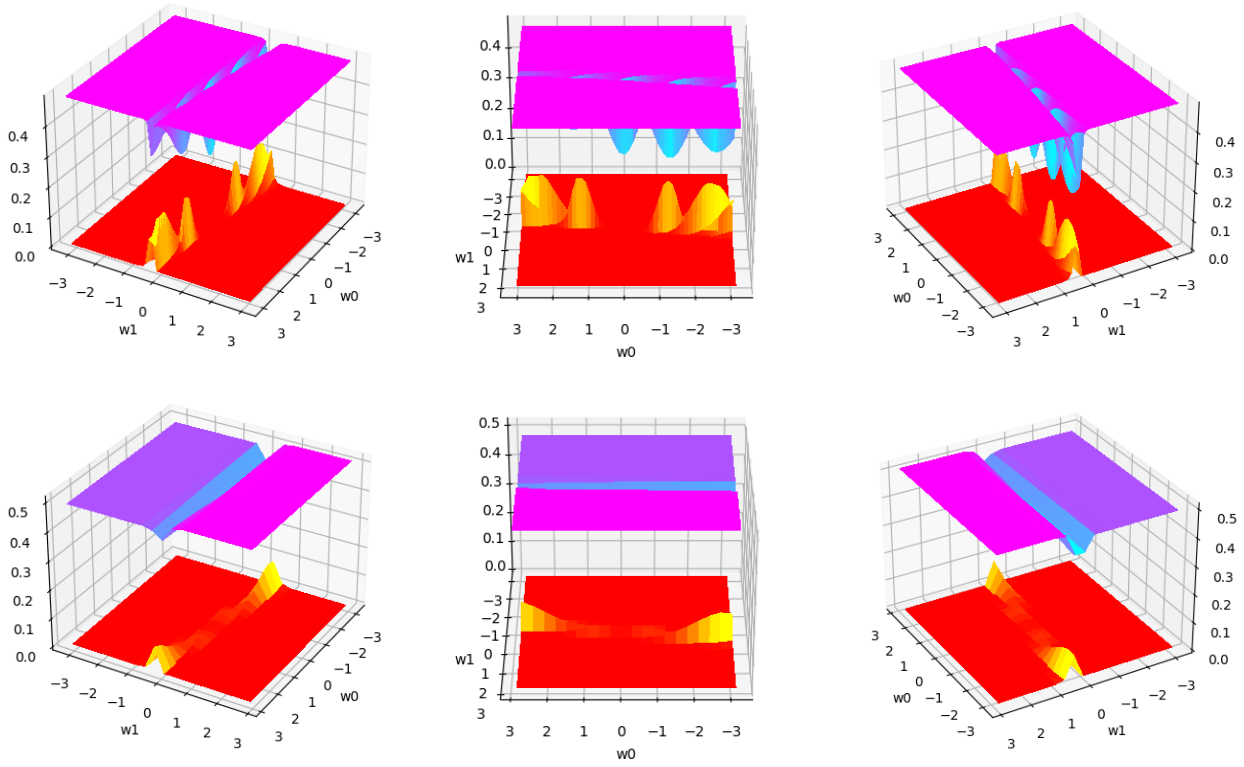


FIGURE 8.2. Fairness-loss landscape for f_1 (top) and f_2 (bottom). Loss values are presented in the cool color gradient and fairness values in autumn. One can see that there are several local minima with respect to loss, each corresponding with a different value for fairness.

TABLE 8.1. Demonstration of our calculated expected value for fairness compared to the actual results.

# Parameters	Expected Fairness	Fairness After Training	Difference	Time For Integral
2	0.1134	0.1107	2.38%	41 minutes
3	0.1348	0.1385	−2.64%	2 hours 28 minutes

While the results calculating the expected value of fairness are close to the actual values, they are not exactly on the mark. We propose a few reasons as to why this may be. First, we assume that our neural network would be trained to convergence, though there in practice neural networks only approach a local minimum, and may not ever reach there exactly.

8.4. Conclusion

While the bias embedded into data has the capacity for harm, the fact of the matter is that algorithmic bias is a product of the intersections between algorithm and data. While existing work has focused primarily on examining the causes of bias from a data perspective, we prove that some of the bias of a neural network can be attributed to the algorithm itself. We also demonstrate that the expected bias for a neural network given a set of hyperparameters and a population over which it will serve is calculable before training. We propose an integral over parameter space to find the expected value of fairness and demonstrate that this integral is very similar to the actual values of unfairness one gets from training the model, though the process is computationally expensive.

We argue that understanding fairness as not only a data issue, but also an algorithm issue is essential for tackling the problem, and we propose a framework for this sort of analysis.

We propose an approximation scheme for larger networks that utilizes the relatively new framework of loss landscapes to approximate this integral. Future work should be directed towards the question of the fidelity of compressing a neural network's parameter space in this way to determine if such an approach is viable.

CHAPTER 9

(Un)fair Backbones In Neural Network

Abstract

Fairness in machine learning is concerned with creating equitable outcomes for different sensitive identities, though there are several limitations to the prevailing ways researchers have attempted to create fair systems. One of these is the lack of consideration of intersections of identity, despite the fact that social scientists have long argued that such intersections are more than the sum of their parts. Here, we extend previous research done on neural network backbones, the idea of mining meaningful subgraphs within a neural network's structure. While previous work was dedicated to mining patterns associated with *misprediction*, here we try to uncover subnetworks associated with *unfair* prediction. We demonstrate that neural networks have distinct patterns for when they are unfairly privileging, acting biased against, or acting fair to, certain groups, and we consider three styles of leveraging this fact to ensure more equitable behavior: an in-situ approach where networks train to become more fair, a post-hoc auditing approach where the unfair network's behavior is monitored with suspicious behavior being flagged, and a zero-shot approach of modifying a network's parameters without training such that the network will behave more fairly. We also examine the implications of these findings to intersections of identity, and demonstrate that our approach can significantly improve fairness with respect to such groups.

9.1. Introduction

Fair machine learning is concerned with ensuring that different protected classes are treated equitably in some downstream task [20]. While this is true for all fair machine learning approaches, techniques can be taxonomized in three different ways [23]: pre-processing techniques which try to introduce fairness before modelling [201], in-processing (or in-situ) techniques which introduce fairness during modelling [202,203], and post-processing (or post-hoc) techniques which are applied after modelling [202,203]. While each of these techniques has found some degree of success for the task they study, one of the major limitations of many fair machine learning approaches is a lack

of consideration to intersections of identity [204]. This is in spite of the fact that social scientists have repeatedly demonstrated that dimensions of identity intersect in meaningful ways [33], ie the experience of a transgender man cannot be modeled by the experience of being a man plus the experience of being transgender in a more abstract sense.

Here, we create approaches which address this issue and stretch across the latter two sections of the aforementioned taxonomy by leveraging deep learning backbones, a technique of mining meaningful subgroups of neurons that are highly associated with a particular user defined concepts and not associated with other concepts [138]. These have been studied with respect to model performance. The original paper [138] examined if commonalities could be mined for how neural networks make mistakes, and it was demonstrated that one could identify, with high precision, when neural network was likely to make a mistake and correct that misprediction. Here, we hope to extend this work for fairness: identifying how and when unfair predictions take place and correcting them - either by modifying individual decisions or the network as a whole.

This paper also extends the novel idea of machine-to-machine explanation. Here, explanations of unfair pathways are not meant to be meaningful to humans, but rather, they are meant to explain to the neural network how unfairness is occurring in their hidden units, and attempting to correct this issue. This paper serves as an exploration into the intersection between machine-to-machine explanations and fair machine learning.

We make the following contributions:

- We demonstrate that the concept of deep learning backbones can be extended to fairness settings, that is, neural networks often create unfair pathways that can be mined.
- We leverage these unfair pathways to create three techniques to alleviate unfairness in the network, those being:
  - A post-hoc technique for identifying individual unfair predictions
  - An in-situ approach for training a fair model via a directed dropout
  - A zero-shot fair machine learning technique in which a network is made more fair without extra training by dropping unfair pathways

We begin by explaining the previous framework of neural network backbones before discussing the three approaches to fairness in more detail. We present our experimental results on the COMPAS dataset [171] and demonstrate how it can be scaled to intersections of identity without much extra computational cost. Finally, we discuss insights gained though this approach and conclude.

9.2. Related Work & Backbones

The idea of neural network backbones is to consider the feed-forward layers of a network to be a graph with edges connecting neurons from one layer to the next which is meaningful to some user defined concept. Concepts are defined by a series of instances which activate when the concept is present, but not for other concepts. For example, to explain the concept of dogs vs cats, the neural network backbone for dogs would be activated when the network sees a dog but not a cat, and the backbone for cats would activate when the network sees a cat but not a dog. Previous work formulates the concept-level (CL) backbone identification problem as finding the minimum sub-network that exhibits:

- **Coverage:** the subgraph covers the activations of the user-defined concept
- **Layer-Inclusion:** layers from the first fully-connected layer to the penultimate layer are included in the subgraph
- **Conected:** Neurons from one layer are connected to the next by a non-zero weight

The last two constraints together are called the "complete graph" constraint. A model-level (ML) backbone is a series of CL-backbones which are distinct from each other. This problem naturally lends itself to being an ILP, though this is proved intractable (NP-Hard), and an efficient, polynomial-time heuristic which uses a novel frequent pattern mining is used to generate the backbones [138].

There is reason to believe that neural network exhibit unfair sub-pathways which can be mined and exploited. The study of adversarial fairness [32, 157] attempts to make networks more fair by training an adversary on the intermediate layers of a neural network to classify an individual as being part of a protected status group. If the adversary can do this, it implies that reasoning is being done over the protected status space, even if the dimensions of identity are not specifically provided to the network. These works train a network not only for their original task, but also

to ensure that the adversary cannot determined protected status from the hidden unit activations, implying that the network is no longer considering such information in its decisions.

This is, in effect, mining the hidden unit activation space to detect unfairness, however it is much less complex than the proposed method, as the input to the adversary is typically a single layer's activations whereas we seek to mine a complete subgraph.

9.3. Approach

In this section, we discuss our approach to unfairness and the novel forms of intervention we will explore.

9.3.1. Backbone Concepts. Here we use the same algorithm where the user defined concepts are groups of people who are being treated unfairly and people who are being treated fairly. This is accomplished by querying the network on different populations and sub-populations based to see which groups are being treated fairly or unfairly. Here, fairness is defined as predictive-equity, that is:

$$P(d = 1|Y = 0, P = a) = P(d = 1|Y = 0, P = b) \tag{9.1}$$

Where d is the predicted value, Y is the label, and P is the protected attribute. That is, a network is considered fair if we mistakenly allow people into the positive class at equal rates [205]. While over 20 definitions of fairness exist, this is one of the most common definitions used in the literature for supervised learning [22], which is the area we are studying in this paper. To get a score of unfairness, we consider the false positive rate (FPR) of the specific population, subtract this value by the FPR of the data overall, and normalize it by the overall FPR:

$$Unfairness(X, Y, p) = \frac{FPR(X, Y|P = p) - FPR(X, Y))}{FPR(X, Y)} \tag{9.2}$$

That way, if a group is treated preferentially, they will have a higher FPR than the general FPR and this value will be positive. If a group is treated with bias, they will have a negative value for unfairness, and if a group is being treated equitably, they will have an unfariness value of 0.

After this query is done, we will know which groups are being treated with the most bias and privilege. The instances inside the populations are labeled the the biased group and the privileged group, respectively. The groups are further sub-divided into fairly treated (ie they are correctly predicted) or unfairly treated (ie they are mispredicted in a way that increases their bias or privilege), and a model-level backbone is created to explain the difference between the unfairly treated biased group and unfairly treated privileged group, and another is created to explain the difference between fairly predicted privileged group and fairly predicted biased group.

9.3.2. Performance Desidrata. In order to evaluate how the different algorithms perform, we have two desired characteristics for the approach.

(1) The model is made more fair than it otherwise would be

(2) The performance of the model is minimally affected

Performance-fairness tradeoffs are very common in fair machine learning [21, 23], though at times a fair intervention may increase performance [32], particularly when the bias signal is greater in the training set than the test set.

We say a technique dominates another if it is better on both criteria, and is pareto efficient if one could not get better results from one criterion without sacrificing another.

Fairness is reported with three numbers for our case study, the COMPAS dataset [171]. We report fairness with respect to gender, averaged across genders, fairness with respect to race averaged across races, and fairness with respect to intersections of gender and race, averaged amongst them.

9.3.3. Post-Hoc Auditing. For the first approach, we apply a technique very similar to the original backbones paper [138]. In this work, both the hidden unit activations of an instance and a network's prediction of that instance are considered in an approach to relabel a prediction. In this, a ML-backbone exists to explain how instances are commonly correctly predicted as a class (ML_+ backbone) and another is created to explain how instances are commonly incorrectly predicted as a class (ML_- backbone). First, the Jaccard similarity between the activated neurons of the instance is compared to each subgraph of the ML_+ backbone. If the most similar backbone is the backbone of the predicted class, the prediction is accepted. If not, it is compared to the subgraphs of the ML_- backbone. If the most similar backbone is that of the predicted class, the prediction is changed,

otherwise it is accepted. Essentially, the pipeline asks the question "does this look like a typical correct prediction of this class?", and then "does this look like a typical mistake?". If the answer to the first question is no and the answer to the second is yes, this is flagged as a mispredition. In that paper, this lead to significant performance increases for a state-of-the-art architecture for bird audio detection.

The extension to fairness is straightforward. Using the same approach, if the prediciton is positive, we ask the question "does this look like a typical fair positive prediction?" if so, it is accepted, otherwise we ask "does this look like a typical unfair positive prediction?" and if it does we flip the label of the prediction. Similarly, if the prediction is negative, we ask the compliment of those questions. We also examine one other pipeline, in which we compare Jaccard similarity of the network's activations to the unfair privileged backbone and fair privileged backbone in the case of a positive prediction, or the unfair biased backbone and fair biased backbone. If the instance looks more like an unfair prediction that a fair prediction, the prediction's label is flipped.

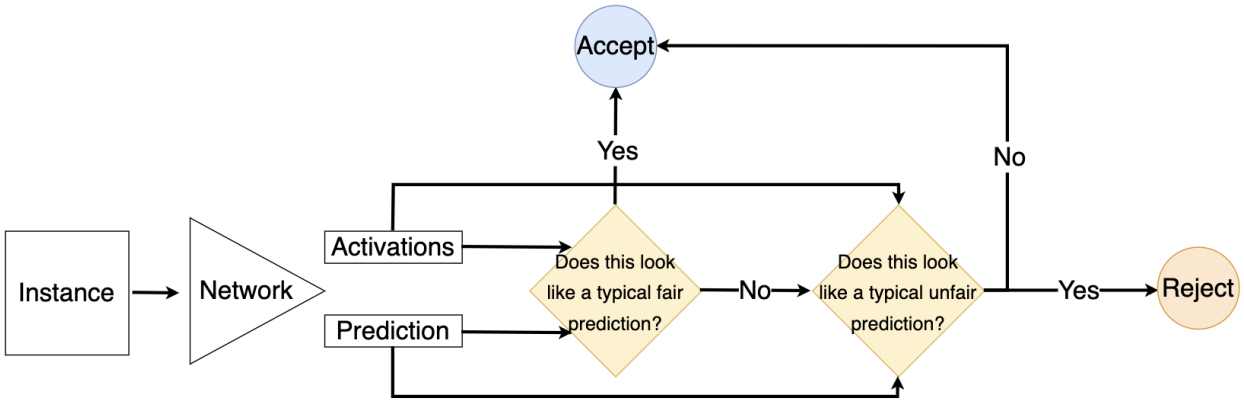


FIGURE 9.1. Pipeline for post-hoc auditing approach. An instance is passed through the network, producing a binary activation vector and a prediction. These are used to see if this looks like a typical fair prediction and not a typical unfair prediction. If the pipeline deems the prediction suspicious, the predicted label is flipped.

9.3.4. In-Situ Directed Dropout. In the next approach, we examine the idea of doing a guided dropout of unfair subnetworks during training. In this approach, we mine backbones for unfair predictions and perform dropout over those neurons during training. This way, as a neural network learns to exploit protected statuses for performance, that learned information is lost before it can be ingrained deeper into the model. We call this directed dropout because

instead of randomly dropping out the influence of certain neurons, we chose neurons by the ones which are most contributing to unfair results.

We also experimented with the idea of moderating this search such that any overlap with the fair network was maintained (ie we cannot dropout neurons if they exist in the fair pathway), though we found that the aforementioned approach dominated this one.

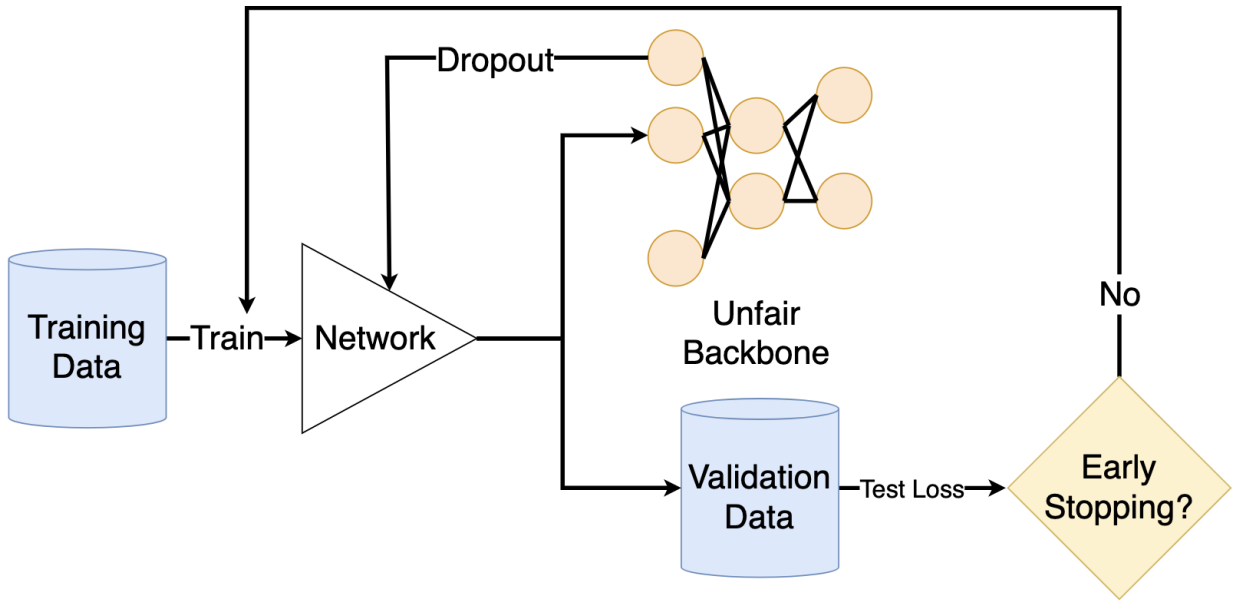


FIGURE 9.2. Pipeline for the in-situ approach. The model is trained, unfair backbones are gathered, and that backbone is dropped out before training continues. Training terminates with early stopping.

9.3.5. Post-Hoc Zero-Shot Learning. In the final approach, we explore a method of modifying a neural network's parameters without training in hopes of making it more fair. In this approach, we train a model as normal, but after training we isolate the unfair subgraphs and dim their influence (multiply their parameters by some factor $\alpha \in [0, 1)$). The hope is that by removing such neurons we will be able to keep the majority of predictive power of the network while removing the neurons responsible for unfairness. This is an application of explainable artificial intelligence for the task of network pruning [206, 207], in which specific neurons are dropped out in the same way to make the network more efficient, compact, or powerful.

We experiment with three different approaches: in the first, we remove all neurons from the unfair biased backbone that are not in the fair biased backbone. In the second, we remove all

neurons from the unfair privileged backbone that are not in the fair privileged backbone. Finally, we remove all neurons which are in both the unfair backbones.

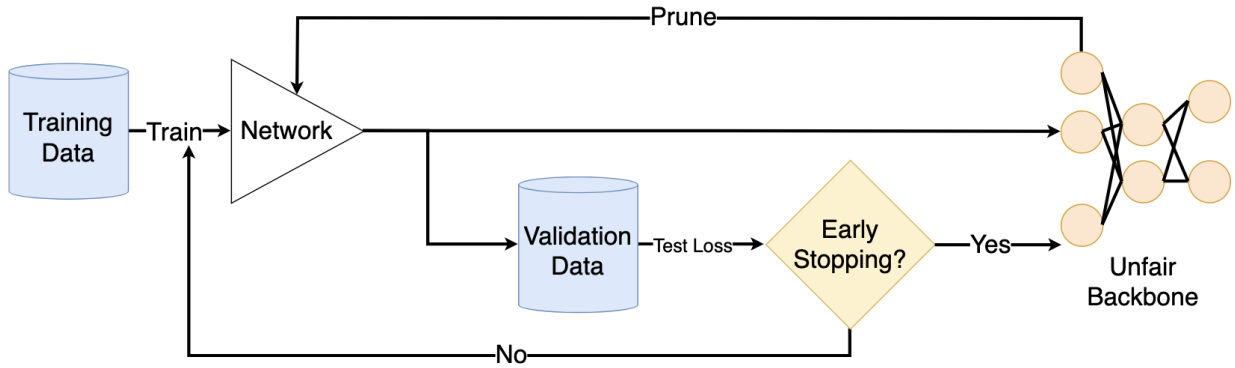


FIGURE 9.3. Pipeline for the zero-shot learning approach. The network is trained normally, unfair neurons are mined, and they are dropped out of the learner.

9.4. Experimental Results

The neural network used for experiments is a two-hidden-layer network with 256 and 128 neurons for the first and second layers, respectively. The network is trained via an Adam optimizer [208] with early stopping if the network does not surpass its best validation set loss within three epochs. Training and validation sets follow an 80-20 split. Networks are trained over five random initialization, with training and validation sets being shuffled at each initialization. Median results for the five initializations are provided. Recall that increases and decreases are percentages of their original (before intervention) strategies, e.g. a change of -50% for unfairness may imply unfairness went from 0.2 to 0.1.

Tables 9.1 and 9.2 provide an overview of how both criteria change relative to their base models. As discussed in the preceding section, we come up with different formulations for these approaches. Table 9.1 shows the approach that led to the fairest outcome, while 9.2 shows the approach that provided the lowest error. As one can see, no approach dominates any other, and each has at least one change better than all others.

The Post-Hoc auditing approach created the fairest models when we compared the activation vector from an instance to both the fair and unfair model and relabeled if it was more similar to the fair model. This created near equity for the racial group and reduced intersectional unfairness by over 75%, however, this was at a much larger error increase (28.07%) than other models. While

some predictive power is maintained, this may be too large of a cost for some people. The auditing approach that yielded the lowest change in accuracy was the pipeline visualized in Figure 9.1. This yielded much more subtle changes to gender unfairness but remained the best change to racial fairness out of the group. Changes to intersectional unfairness were also quite large at -21.07%, with a negligible increase in error of only +0.2026%.

The in-situ technique tended always to increase fairness, though less so than the other techniques. Interestingly, it consistently decreased error when implemented, with a median decrease of -1.28%. The in-situ strategy that worked the best was dropping out all unfair neurons, and this technique dominated all others listed in the preceding section. Notably, the in-situ technique also comes at the cost of increased training time, as new pathways must be mined each epoch. In our experiments, they added six minutes to every epoch on average (the training time takes less than one second per epoch, though finding deep learning backbones is not directly dependent on training time. an $O(k * n^2)$ process where n is the number of transactions and k is the number of neurons).

Finally, the zero-shot learning approach also made the model significantly more fair with respect to all metrics and a relatively low cost of error for fairness gained. The most fair version of this approach was muting all neurons associated with unfairness (privileged and biased). This dropped a comparable amount of racial and intersectional unfairness to the post-hoc auditing technique in the same group but at a significantly less increase in error, allowing the model to become fairer while maintaining the vast majority of the predictive power of the model. Further, if this error cost is deemed too large by the user, this is the only technique that allows the user to moderate how much error they are willing to gain for the amount of increased fairness they receive. Visualized in Figure 9.4, one can see that error tends to increase with increasing values of α, while unfairness for all aspects of the model tends to decrease. In this model, if the increase of error by 15.63% is too steep for the user, they could walk this back to more moderate changes.

The version of this approach that yielded the minimal increase in error was muting the privileged backbone. This increased error by a mere 3.99% while decreasing intersectional unfairness by 64.22%.

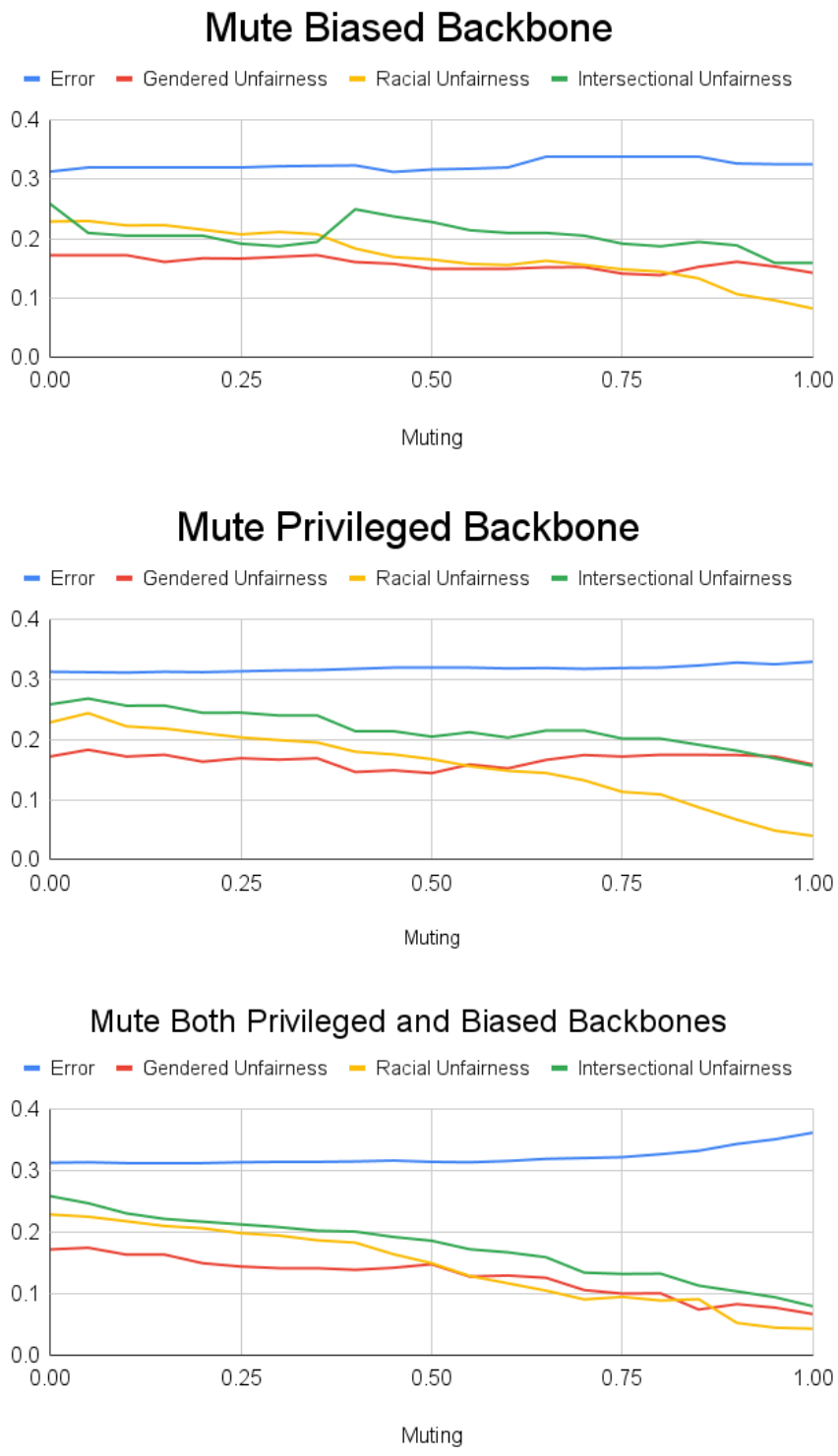


FIGURE 9.4. Results of the zero-shot learning approach with varying levels of muting. Typically, unfairness goes down the more unfair pathways are muted, and performance may also slightly degrade with higher values of α.

TABLE 9.1. Changes in fairness and error for the different approaches, favoring more fair models. The best metric for each criterion is in bold.

Method	Change in Gender Unfairness	Change in Racial Unfairness	Change in Intersectional Unfairness	Change in Error
Post-Hoc Auditing	-31.89%	**-92.4%**	**-75.45%**	+28.07%
In-Situ Dropout	-23.14%	-8.561%	-13.00%	**-1.28%**
Post-Hoc Zero-Shot	**-61.13%**	-81.19%	-69.28%	+15.63%

TABLE 9.2. Changes in fairness and error for the different approaches, favoring more performant models. The best metric for each criterion is in bold.

Method	Change in Gender Unfairness	Change in Racial Unfairness	Change in Intersectional Unfairness	Change in Error
Post-Hoc Auditing	-2.207%	**-75.64%**	-21.07%	+0.020%
In-Situ Dropout	-23.14%	-8.561%	-13.00%	**-1.28%**
Post-Hoc Zero-Shot	**-38.62%**	-17.33%	**-64.22%**	+3.99%

9.5. Conclusion

Our results demonstrate that there may exist unfair pathways in trained neural networks that may be reasoning over the protected status-space. These patterns can be mined, and that information can be used in a variety of ways to increase the fairness of a model. In this paper, we examined three such approaches: a post-hoc auditing approach where the activation of a test-set instance is compared to different backbones and the prediction of the model can be overridden if the activations seem unfair, an in-situ approach in which pathways associated with unfairness are dropped after each epoch, and a post-hoc zero-shot approach in which these neurons are dropped only after training.

Each technique excelled in different areas, though in our opinion the post-hoc zero-shot approach yields the highest return for the error gained. Further, it is the only technique to include a parameter to let the user decide how much of an increase in error is worth in terms of extra fairness. Another

technique of note is in-situ dropout which not only made the network more fair (though, by more moderate degree than all other techniques), but also decreased error.

Ultimately, this serves as a final example of machine-to-machine explanation and demonstrates a meaningful intersection between explanation techniques and fair machine learning.

CHAPTER 10

Conclusion

This dissertation demonstrates the power of the novel concept of machine-to-machine explanation - leveraging explainable artificial intelligence for tasks beyond explaining a model to a human. Specifically, we have demonstrated that counterfactual explanations can be used to embed learned knowledge from one model to another (Chapter 2), our novel explainable anomaly detection algorithm can create virtual instances for self-supervised learning (Chapter 3), our explainable clustering approach can be used for instance transfer learning (Chapter 4), and our formulation of deep learning backbones can be used to predict errors and unfair predictions.

We also demonstrate that the existing body of fair machine learning has several key limitations, namely, that most techniques in fair machine learning do not consider the intersections of identity as anything beyond two disentangled variables, while the research in social science indicates that such intersections are meaningful [33]. Some of our work in machine-to-machine explanation demonstrates techniques capable of handling intersections (Chapter 9). Further, one of the prevailing attitudes is that fairness is, at its core, a data problem that can be addressed when or after training algorithms. We demonstrate, however, that the fairness problem is a complex intersection between algorithm and data. Specifically, we examine the causes of unfairness and determine that while some may be wholly data problems (eg sample size bias and labelling noise), others are interactions between the data and algorithm (eg incompressibility) (Chapter 7). Finally, we also mathematically prove and empirically demonstrate that some of the bias of a neural network can be attributed to the model architecture and hyperparameters alone (Chapter 8).

I hope that these insights will help guide future research to recognize the potential for XAI beyond explanation to humans and that future research will be dedicated to adding to these approaches which address major limitation in the existing work.

APPENDIX A

An Exemplars-Base Approach for Explainable Clustering: Complexity and Efficient Approximation Algorithms - Proofs, Runtimes, & Exemplars

A.1. Additional Material for Section 4.3

Here, we present definitions of some graph theoretic concepts and combinatorial problems used in our work.

Graph Theoretic Definitions: We use some graph theoretic concepts and a special class of graphs in proving our results. Given an undirected graph $G(V, E)$, a subset V' of nodes forms a **dominating set** for G if for every node $w \in V - V'$, there is a node $v \in V'$ such that the edge $\{v, w\}$ is in E. Given a graph $G(V, E)$, the goal of the **minimum dominating set** (MDS) problem is to find a dominating set of minimum cardinality for G.

Given a set of disks (i.e., circles in two-dimensional space) each with the same radius r, one can define an associated undirected graph as follows: there is one node for each disk; there is an edge between two nodes if the corresponding disks touch or intersect (i.e., the distance between the centers of the disks is at most $2r$). Such a graph is called a **unit disk graph** [209]. Many optimization problems, including the MDS problem, are known to be NP-hard even for unit disk graphs [127, 209]. We rely on the NP-hardness of the MDS problem for unit disk graphs in proving Theorem 4.4.1.

Unit disk graphs can be defined in three or more dimensions where each object is a ball of unit radius in an appropriate dimension. Each node of the corresponding graph represents a ball with an edge between two nodes if their corresponding balls touch/intersect.

Minimum Set Cover (MSC) Problem: In this problem [123], the input consists of a base set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $Y = \{Y_1, Y_2, \ldots, Y_m\}$, where each Y_j is a subset of U $(1 \leq j \leq m)$ and an integer bound $\beta \leq m$. The goal is to choose a subcollection Y' of Y with $|Y'| \leq \beta$ such

that the union of the sets in Y' is equal to U (i.e., the union covers all the elements in U). This problem is NP-complete and a natural greedy approximation algorithm (which picks a new set in each iteration such that the set covers as many new elements as possible) is known to give a performance guarantee of $O(\log n)$ for the problem [124]. One of our results (Section 4.4.3) uses this approximation algorithm.

Budgeted Maximum Coverage Problem: We also use a known approximation algorithm for the Budgeted Maximum Coverage (BMC) problem, which is closely related to the Minimum Set Cover (MSC) problem [123]. The input to the BMC problem is a base set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $Y = \{Y_1, Y_2, \ldots, Y_m\}$, where each Y_j is a subset of U $(1 \leq j \leq m)$ and a budget $\beta \leq m$. The goal is to choose a subcollection Y' of Y with $|Y'| = \beta$ such that the union of the sets in Y' covers the maximum number of elements of U. This problem is also NP-hard and a natural greedy approximation algorithm (which picks a new set in each iteration such that the set covers as many new elements as possible) has been shown to give a performance guarantee of $(1 - 1/\mathrm{e})$ for the problem [126], with e being the base of the natural logarithm. One of our results (Section 4.4.3) uses this result.

A.2. Additional Material for Section 4.4

A.2.1. Statement and Proof of Proposition 4.4.1. Statement of Proposition 4.4.1: The MSE problem is NP-hard even when the set of instances X consists of points in two-dimensional Euclidean space and the distance between any two points is their Euclidean distance.

Proof: The proof is by a straightforward reduction from the minimum dominating set (MDS) problem for unit disk graphs discussed in Section A.1. Let the MDS problem be specified by a unit disk graph $G(V, E)$, where the radius of each disk is r, and let $\beta \leq |V|$ be the given upper bound on the size of a dominating set. We construct a set of instances X for the MSE problem as follows. For the disk corresponding to each vertex v_i, we create an instance $x_i \in X$, where the coordinates of x_i are those of the center of the disk corresponding to v_i. The exemplar distance ϵ is set to $2r$ and the bound on the number of exemplars is set to β. Obviously, this construction can be done in polynomial time.

Suppose V' is a dominating set for G with at most β nodes. We can show that the instances corresponding to the nodes in V' form the exemplar set $\mathcal{E}$ for X as follows. Consider any instance x_j in X which is not an exemplar. Since V' is a dominating set and the node v_j corresponding to x_j is not in V', there is a node $v_i \in V'$ such that the edge $\{v_i, v_j\}$ is in E. Since G is a unit disk graph, the distance between the centers of the disks corresponding to v_i and v_j is at most $2r$ which is equal to ϵ by our construction; that is, the distance between x_j and the exemplar x_i is at most ϵ. Therefore, $\mathcal{E}$ is a set of exemplars of size at most β for X.

Now, suppose $\mathcal{E}$ is a set of exemplars of size at most β for X. Let V' be the set of nodes of G corresponding to the instances in $\mathcal{E}$. We claim that V' is a dominating set for G. To see this, consider any node v_j which is not in V'. The instance x_j corresponding to v_j has an exemplar $x_i \in \mathcal{E}$ and the distance between x_i and x_j is at most $2r$. Since G is a unit disk graph, the edge $\{v_i, v_j\}$ is in E. In other words, V' is a dominating set for G, and this completes the proof. QED

A.2.2. Statement and Proof of Theorem 4.4.2. Statement of Theorem 4.4.2: The solution produced by Algorithm 2 satisfies the following conditions: (i) The diameter of each cluster is at most $2(D^* + \epsilon)$, where D^* is the optimal diameter for a k-clustering of X and ϵ is the exemplar distance. (ii) Every instance in X has an exemplar (at a distance of at most ϵ) within the same cluster. (iii) The sets of exemplars for the k clusters are pairwise disjoint. (iv) The total number of exemplars generated by the algorithm is at most $O(N^* \log n)$, where N^* is the minimum number of exemplars needed to cover all the instances in X.

Proof: To prove Part (i), we first note that the approximation algorithm used in Step 1 guarantees that the maximum diameter of the clusters produced in that step is at most $2D^*$, where D^* is the optimal solution value for X. Step 6 of the algorithm moves only non-exemplars between clusters. We need to show that after these moves, the maximum diameter is at most $2(D^* + \epsilon)$. To see this, consider any cluster C_i and any pair of instances x_a and x_b in C_i. There are three cases to consider.

Case 1: Both x_a and x_b are exemplars. In this case, both x_a and x_b must be in B_i since we chose $\mathcal{E}_i = B_i \cap A$. Thus, at the end of Step 1, $d(x_a, x_b) \leq 2D^*$.

Case 2: One of them, say x_a, is an exemplar and the other (i.e., x_b) is a non-exemplar that got moved into C_i. In this case, C_i contains an exemplar x_q at a distance of at most ϵ from x_b. Since $d(x_a, x_q) \leq 2D^*$ and $d(x_q, x_b) \leq \epsilon$, it follows from triangle inequality that $d(x_a, x_b) \leq 2D^* + \epsilon$.

Case 3: Both x_a and x_b are non-exemplars which were moved into C_i. In this case, C_i contains exemplars x_p and x_q such that $d(x_a, x_p) \leq \epsilon$ and $d(x_b, x_q) \leq \epsilon$. Further, $d(x_p, x_q) \leq 2D^*$. Now, using triangle inequality, it follows that $d(x_a, x_b) \leq 2(D^* + \epsilon)$, and this completes our proof of Part (i).

The result in Part (ii) follows since the set A constructed in Step 3 is an exemplar set for X and each non-exemplar instance x_j gets moved (in Step 6) to a cluster containing an exemplar for x_j. Since the blocks constructed in Step 1 are pairwise disjoint, so are the exemplar sets constructed in Step 5; this proves Part (iii). Since Step 3 uses the greedy approximation algorithm for MSC and this algorithm provides a performance guarantee of $O(\log n)$ [124], the total number of exemplars produced in Step 3 is at most $O(N^* \log n)$, where N^* is the minimum number of exemplars needed to cover all the instances in X. This establishes Part (iv) and the theorem follows. QED

Expanded version of the Remark in Section 4.4.2: The remark in Section 4.4.2 mentions that one can theoretically get a better performance guarantee for the number of exemplars chosen by Algorithm 2. Here, we explain how such an improvement can be obtained.

Since Step 3 in Algorithm 2 uses an approximation algorithm for MSC, the performance guarantee with respect to the number of exemplars is $O(\log n)$, where $n = |X|$. Theoretically, one can get a better approximation by transforming the Exemplar Selection steps (i.e., Steps 2 and 3 of the algorithm) into that of finding a near-optimal dominating set for unit disk graphs in an Euclidean space whose dimension ℓ is the same as that of the points in X. This is done by placing an ℓ-dimensional ball of radius $\epsilon/2$ at each instance in X. The corresponding unit disk graph has a node for each instance in X and there is an edge between two nodes if the corresponding balls intersect or touch. It can be verified that any dominating set for this graph provides the necessary set of exemplars. An approximation scheme which provides a performance guarantee of $(1 + \delta)$ for any fixed $\delta > 0$ is known for the minimum dominating set problem for such graphs [127]. Thus, one can obtain a performance guarantee of $(1 + \delta)$ for any fixed $\delta > 0$ with respect to the number of exemplars. However, this approximation scheme is impractical even for data sets of moderate size since its running time has the factor $O(n^{(1/\delta)^2})$. (Thus, even when $\delta = 0.5$, the running time has the factor $O(n^4)$.) For this reason, we decided to use the MSC-based approximation algorithm in our experiments.

Running time of Algorithm 2: We can estimate the asymptotic running time this approximation algorithm as follows. Step 1 uses Gonzalez's algorithm which has a running time of $O(nk)$, where n is the number of instances and k is the number of clusters [122]. Step 2 constructs the neighborhood set for each instance and can be done in time $O(n^2)$. Step 3 runs the greedy set cover heuristic for which the running time is $O(W)$, where W is the sum of the sizes of all the sets [210]. In our case, since there are n sets and each set is of size at most n, $W \leq n^2$; that is, Step 3 runs in time $O(n^2)$. Step 4 runs in $O(k)$ time. Using a bit vector representation for each set, Steps 5 and 6 can be implemented to run in time $O(nk)$. Since $k \leq n$, the overall running time of Algorithm 2 is $O(n^2)$.

A.2.3. Statement and Proof Theorem 4.4.3. Statement of Theorem 4.4.3; The solution produced by Algorithm 3 satisfies the following properties: (i) The diameter of each cluster is at most $2(D^* + \epsilon)$, where D^* is the optimal diameter for a k-clustering of X and ϵ is the exemplar distance. (ii) The sets of exemplars for the k clusters are pairwise disjoint. (iii) The total number of instances with exemplars is at least $(1 - 1/\mathrm{e})Q^*$, where e is the base of the natural logarithm and Q^* is the maximum number of instances in X that can have exemplars under the constraint that the total number of exemplars is at most β.

Proof: The proofs of Parts (i) and (ii) are identical to the ones given in the proof of Theorem 4.4.2. Part (iii) follows from [126] that the greedy approximation algorithm for BMC covers at least $(1 - 1/\mathrm{e})Q^*$ elements, where Q^* is the maximum number of elements that can be covered using at most β sets. QED

Running time of Algorithm 3: The estimation of the asymptotic running time of Algorithm 3 is similar to that of Algorithm 2. The main difference between the two algorithms is that while Algorithm 3 uses the greedy algorithm for the BMC problem in Step 3 while Algorithm 2 uses the greedy algorithm for the Minimum Set Cover (MSC) problem. However, the asymptotic running time of the greedy algorithm for BMC is also the same as that of the greedy algorithm for MSC [126]. Therefore, the running time of Algorithm 3 is also $O(n^2)$.

A.3. Additional Material for Section 4.5

A.3.1. Time Complexity. Our approximation algorithms run in polynomial time (more precisely, in $O(n^2)$ time in the worst-case) and have strong performance guarantees in terms of clustering quality and explanation complexity. The run times for our algorithms are as expected not as fast as simple k-means style algorithms but our work comes with performance guarantees with respect to optimal solutions and are much faster than state of the art domain specific methods. For example in our work on explaining deep embeddings for text (Section 4.5.2), our SCCE and SCCRB algorithms took 93 and 96 seconds respectively whilst the state of the art method took 700+ seconds and k-means style algorithms (which lack explanation) took under 10 seconds. Our algorithm has just two parameters, namely k and ϵ, where the latter parameter naturally trades off clustering quality and explanation complexity.

A.3.2. Experimental Details of Harry Potter Explanation Experiments. We represent each sentence in the first HP book using the state-of-the-art language model BERT [133]. Hence, the exemplars generated by our method will be sentences in the book. Specifically, we fine-tune a pre-trained BERT-base model (`https://huggingface.co/`) in two steps. First, we add to the vocabulary terms words that are unique to the Harry Potter universe (e.g., "quidditch") and train the model with a very low learning rate. Then, we fine-tune the model to produce a relevant sentence embedding using the Sentence-BERT architecture [211] to create a HP Specific BERT model. *It is important to note that all methods and baselines use this embedding scheme.*

A.3.3. Harry Potter Explanations By Our Method. Here we present the explanation generated by our approach. We color code the exemplars by the cluster they belong to.

At that moment the telephone rang and Aunt Petunia went to answer it while Harry and Uncle Vernon watched Dudley unwrap the racing bike a video camera a remote control airplane sixteen new computer games and a VCR. One small hand closed on the letter beside him and he slept on not knowing he was special not knowing he was famous not knowing he would be woken in a few hours' time by Mrs Dursley's scream as she opened the front door to put out the milk bottles nor that he would spend the next few weeks being prodded and pinched by his cousin Dudley. Harry didn't sleep all night. Perhaps it was because he was now so busy what with Quidditch

practice three evenings a week on top of all his homework but Harry could hardly believe it when he realized that he'd already been at Hogwarts two months. Don' mention it said Hagrid gruffly. Hagrid grinned at Harry. I was allowed ter do a bit ter follow yeh an' get yer letters to yeh an' stuff. There was only one room inside. he leapt to his feet and ran to the window. It got to its feet and came swiftly toward Harry. But he couldn't do it. He sat up and felt around his eyes not used to the gloom. But he never wanted you dead. Hermione had now started making study schedules for Harry and Ron too. The Chasers throw the Quaffle and put it through the hoops to score Harry recited

A.3.4. Quantitative Experiments on Facial Data - Details. We make the task challenging by choosing three similar men (Gerhard Schröder, Jacques Chirac and Tony Blair) and use just 40 images of each person, with half used for clustering and half for testing.

FIGURE A.1. Faces in the Wild Experiments. Exemplars found for our three clusters correspond to the three people used in this experiment, Gerhard Schröder (left), Jacques Chirac (middle) and Tony Blair (right). Note the exemplars of the same person differ mainly by the position of the mouth.

For reproducibility, we simulate a person by the most simple learning algorithm, namely k-nearest neighbor (k-NN). We cluster images of three well-represented individuals from the Labeled Faces in the Wild dataset [212] using our method. Images are first processed into embeddings via FaceNet, a deep embedding network. After clustering using our method, each cluster was assigned the label of its most well-represented individual.

We created three baselines to predict the person in the hold out image: 1) Using a nearest centroid approach, 2) Using a k-NN approach with all points and 3) Using a k-NN approach but with random 20% of points from each cluster. After conducting this experiment five times with five different training/testing splits, we obtained the results summarized in Table 4.3. This experiment demonstrates that exemplars produced by our method are more useful than **other artifacts** of the very same clustering namely centroids, all points and random subsets of points. A possible reason for the improvement is that our method chooses a more diverse collection of instances (Figure A.1).

A.4. Additional Material for Section 4.6

Comparison to DBSCAN and Other Density Based Clustering Methods. Superficially, our method may seem to be similar to DBSCAN [136] and other similar algorithms as it uses notions such as ϵ-neighbors. However, there are several fundamental differences. Firstly, our method is guaranteed to use the specified number or near-minimum number of exemplars, where as DBSCAN, while being a very useful method, does not provide such guarantees. Similarly, our method has an explicit clustering objective (i.e., to minimize the maximum cluster diameter) where as DBSCAN does not. Finally, DBSCAN is a not designed so that the core points can be considered explanations of the clusters. As a consequence, it is not meaningful to compare our method with DBSCAN.

Comparison to Multiple Centroid Methods. An area that is superficially similar to our own work is finding multiple centroids per cluster; these centroids are sometimes referred to exemplars. However, there are significant differences with respect to the definition of an exemplar, the purpose of the exemplars and the efficiency of the algorithms.

The multi-centroid/exemplar methods are specifically focused on identifying multiple centroids in each cluster, where each centroid specifies a new sub-cluster (e.g., [120, 137]). While these methods allow a user to specify the number of clusters k, the algorithms may find more clusters, that is, possible sub-clusters within each cluster [120]. One can view these are finding a one layer hierarchy within each cluster and experimental results typically compare these algorithms against hierarchical clustering methods.

In our work, an exemplar has a very precise definition: namely a point x is an exemplar for another point y if and only if x is within a certain distance from y. The work on multiple centroid clustering has no such definition. Further, the exemplars generated by our methods are motivated by the need to explain clusters rather than to identify sub-clusters and hence yield fundamentally different results. As an illustrative example, consider a cluster with points uniformly distributed throughout it. Methods such as MEAP and K-MEAP [120, 137] will return just one exemplar for the entire cluster, as there are no distinct sub-clusters. However, our methods will return multiple exemplars when ϵ is small enough. Figure 4.3 provides such an example where the clusters are tightly defined with no sub-clusters. Finally, while the methods in [120, 137] provide no formal

performance guarantees with respect to either of the two objectives considered in our work (i.e., the cluster quality and the number of exemplars chosen), our methods have provable performance guarantees for both of the objectives.

APPENDIX B

Identification & uses of Deep Learning Backbones - Proofs

B.1. Proof of Intractability.

We now show that finding a satisfying assignment for the Concept-Level backbone Problem (hence the more complex second formulation) is intractable. Hence, no exact solution can be found in polynomial time for the optimization variant of the problem. This motivates the need for a heuristic solution which we sketch in the next section. The remainder of this section can be skipped on first reading of this paper.

THEOREM B.1.1. *The CL-backbone problem is NP-complete even when the number of categories/class is just 2 and the number of node activations per instance is at most 3.*

Proof: Membership in NP is obvious. We prove NP-hardness through a reduction from 3SAT, an NP-Complete problem [213]. Let x_1, x_2, ..., x_l denote the l variables and Y_1, Y_2, ..., Y_m denote the m clauses of the 3SAT instance. The reduction to the CL-backbone problem is as follows.

: (a) For each variable x_i, we create two tags, denoted by a_i and b_i. (a_i and b_i correspond to the positive and negative literals of x_i). So, the node activation set $N = \{a_1, a_2, \ldots, a_l, b_1, b_2, \ldots, b_l\}$, and $|N| = 2l$.

: (b) For each variable x_i, we create an item s_i with node activation set $n_i = \{a_i, b_i\}$, $1 \leq i \leq l$. (Thus, $|n_i| = 2$, $1 \leq i \leq l$.) Items s_1, s_2, ..., s_l constitute concept C_1.

: (c) For each clause Y_j, we create an item s_{l+j}, $1 \leq j \leq m$. Suppose Y_j contain literals x_{j_1}, x_{j_2} and x_{j_3}. For each literal x_{j_ℓ} in Y_j, if x_{j_ℓ} corresponds to positive literal x_i, then n_{l+j} contains a_i and if x_{j_ℓ} corresponds to the negative literal $\overline{x_i}$, then n_{l+j} contains b_i. (Thus, $|n_{l+j}| = 3$, $1 \leq j \leq m$.) Items s_{l+1}, s_{l+2}, ..., s_{l+m} constitute category C_2.

: (d) The set of items $S = \{s_1, s_2, \ldots, s_{l+m}\}$.

It can be seen that the tag set of each item produced by the above construction is of size at most three.

Suppose there is a solution to the 3SAT instance. we construct tag sets N_1 and N_2 for categories C_1 and C_2 as follows. For $1 \leq i \leq l$, if the given satisfying assignment sets variable x_i to $true$, we add a_i to C_2 and b_i to C_1; if the given satisfying assignment sets variable x_i to $true$, we add b_i to C_2 and a_i to C_1. It can be seen that C_1 and C_2 are disjoint. Since the truth assignment satisfies all the clauses, C_2 has at least one item from each tag set c_{l+j}, $1 \leq j \leq m$. So, C_1 and C_2 constitute a solution to the CLB problem.

Now suppose that there is a solution to the CLB problem. We have the following claim.

Claim 1: For each i, $1 \leq i \leq l$, C_2 contains at most one of a_i and b_i.
Proof of Claim 1: The proof is by contradiction. Suppose for some i, $1 \leq i \leq l$, C_2 contains both a_i and b_i. Note that C_1 contains the item s_i whose tag set is $\{a_i, b_i\}$. Thus, C_1 must contain at least one of a_i and b_i. Now, since C_1 contains both a_i and b_i, we conclude that C_1 and C_2 are not disjoint. This contradicts the assumption that we have a valid solution to the CLB problem, and Claim 1 follows.

Given a solution to CLB, we construct a solution to SAT as follows. Consider each variable x_i, $1 \leq i \leq l$. If tag $a_i \in C_2$, set x_i to $true$. If $b_i \in C_2$ or neither a_i nor b_i appears in C_2, set x_i to $false$. We claim that this is a valid satisfying assignment. First, using Claim 1, it is seen that each variable is set to either $true$ or $false$. Consider any clause C_j. C_2 contains at least one of the tags from n_{l+j}, the tag set of item s_{l+j} corresponding to C_j. Thus, the chosen assignment sets at least one of the literals in C_j to $true$; that is, the clause is satisfied. This completes the proof of Theorem B.1.1. QED

B.2. Proof of Tight Bounds of Algorithm

While the above algorithms overcome the issue of intractability, it is not immediately obvious how it relates to the objective and constraints of the problem. We prove that our approach will always find a non-trivial (not extreme) valid solution to the problem that is pareto optimal to three functions: the objective (minimizing backbone size), minimizing the coverage relaxation, and minimizing the diversity relaxation. Formally:

THEOREM B.2.1. *The graph returned by our algorithm is guaranteed to be a non-trivial Pareto optimal in respect to the Connected collective backbone Problem objective, and minimizing the two relaxations.*

The knowledge of non-trivial Pareto optimality is an important distinction, as it ensures that the returned solution is one of the best possible, knowledge of where on the Pareto surface the answer will lie remains a pressing practical concern. To address this, we allow an optional minimum coverage term λ to ensure that the final graph covers at least a proportion λ of total instances.

B.3. Tight Bounds on Performance of Algorithm

We now demonstrate that our heuristic approximation to the original ILP is pareto optimal in respect to minimizing the two relaxations of the ILP. The following theorem is composed of two lemmas.

Theorem 2 *The graph returned by our algorithm is guaranteed to be a non-trivial pareto optimal in respect to the ILP's objective, and minimizing the two relaxations.*

The proof is based on two lemmas shown below. The first shows that our algorithm maximizes the F-Score whilst the second shows that a maximal F-Score is guranteed to produce a non-trivial (non-extreme) point on the Pareto front of minimizing explanation (f), maximizing coverage (g) and minimizing diversity (h).

Lemma 1: By iteratively adding frequent subgraphs to the solution, our algorithm finds a valid solution to the ILP that maximizes F-Score.

Note that this is non-trivial, as our algorithm's termination condition is that the change in F-Score by the addition of (another) node(s) be negative, and does not directly test for a solution which provides the maximum F-Score, as that is intractable.

For the following proof of lemma 1, we use the following notation:

Let n_i denote the subgraph added to the solution with support m_i for iteration i, and let t be the iteration of the solution which immediately before that which causes the termination condition. Further, allow TP_i, FP_i, FN_i, and F_i to be the true positive rate, false positive rate, false negative rate, and F score for iteration i. First, all $F_t \geq F_{t-o}$ for any positive o by definition of the termination condition, as the algorithm terminates the first time F-score decreases from one

iteration to the next. Further, the validity of the solution returned by the algorithm is trivial, as all subgraphs added to the solution must meet the same validity requirement as the ILP.

This is a proof by induction that F_t ¿ F_{t+o}, where o is any positive number, or, more simply, that F_t is the largest possible F-Score given the data.

Inductive Base Case

The inductive base case is trivial as the algorithm will terminate when the change in F-Score from one iteration to the next is negative. Therefore, F_t is strictly greater than F_{t+1} by definition.

Inductive Hypothesis

$F_i \geq F_{i+1}$ for $i \geq t$

Proof By Induction

The addition of a new subgraph n_{i+1} implies that that $TP_{i+1} = TP_i + |n_i| * m_i$, as $|n_i|$ neurons which were previously not part of the graph are added, and a proportion m_i of instances in the data have that neuron activated. Further, $FP_{i+1} = FP_i + |n_i| * (1 - m_i)$, as a proportion $1 - m_i$ instances do not have the subgraph n_i. Finally, $FN_{i+1} = FN_i - |n_i| * m_i$, as a proportion m_i of instances which had the neurons n_i are now covered. Therefore, given that:

$$F_i = 2\left(\frac{\left(\frac{TP_i}{TP_i+FP_i}\right)*\left(\frac{TP_i}{TP_i+FN_i}\right)}{\left(\frac{TP_i}{TP_i+FP_i}\right)+\left(\frac{TP_i}{TP_i+FN_i}\right)}\right) = \left(\frac{2TP_i}{2TP_i+FP_i+FN_i}\right)$$

We can express F_{i+1} as:

$$F_{i+1} = \left(\frac{2*(TP_i+|n_i|*m_i)}{2*(TP_i+|n_i|*m_i)+(FP_i+|n_i|*(1-m_i)+(FN_i-|n_i|*m_i)}\right) =$$

$$\left(\frac{2TP_i+2|n_i|*m_i}{2*TP_i+2*|n_i|*m_i+FP_t+|n_i|-|n_i|*m_i+FN_t-|n_i|*m_i}\right) = \left(\frac{2TP_i+2|n_i|*m_i}{2*TP_t+FP_t+FN_t+|n_i|}\right)$$

And therefore:

$$F_{i+2} = \left(\frac{2TP_i+2|n_{i+1}|*m_{i+1}+2|n_{i+2}|}{2*TP_t+FP_t+FN_t+|n_{i+1}|+|n_{i+2}|}\right)$$

Finally, since $m_{i+2} < m_{i+1}$ this implies that $F_{i+2} < F_{i+1}$.

Therefore, $F_t \geq F_j$ for any $j \geq t$, and F_t is a valid solution.

This concludes the proof of lemma 1.

Lemma 2 The maximization of F-Score implies a pareto optimal result to the ILP's (cite number) objective (f), the coverage constraint (g) and the diversity constraint (h).

Proof Of Lemma 2

We now discuss how this implies Pareto optimality in respect to the original problem's objective and the minimization of both of the relaxations. We formulate this as a series of three proofs by contradiction demonstrating that by increasing the original problem's objective, tightening the coverage relaxation, or tightening the separability relaxation can only occur at the cost of one of the other items.

- 1. Assume, for contradiction, that the graph can be made smaller without sacrificing coverage or diversity. This implies that there exist a neuron or neurons which can be removed and not disturb coverage. However, since each neuron is frequent, the removal of a neuron necessarily implies a decrease in coverage.
- 2. Assume, for contradiction, that coverage could be greater without increasing the number of neurons or decreasing diversity. This implies that replacing a neuron or neurons with different neurons would increase coverage more than the neurons in the solution. However this is not possible since the most frequent neurons are added first, which means that the least frequent neuron in the solution provides greater coverage than the most frequent neuron not in the solution, which implies that the only way to increase coverage is to increase the number of neurons.
- 3. Assume, for contradiction, that diversity can be increased without decreasing coverage or adding neurons. This implies that there exists some neuron or neurons which exist in two or more CL-Summaries which can be replaced by an equal number of neurons in those summaries and will not decrease coverage. However, since the added subgraphs maximally increase coverage, as established above, this could only be accomplished by adding more neurons than the backbone initially had, which is a contradiction.

This concludes the proof of this Lemma.

By proving the above two lemmas, we have proved Theorem 2. QED

APPENDIX C

Foundations for Unfairness in Anomaly Detection - Case Studies in Facial Imaging Data - Model Details & Raw Data

C.1. Models

Both the AE and SVDD models use the same architecture, and this architecture is modeled off those in [167]. The architecture is summarized below.

Datasets are in a random (reset for each initialization) 80-20 split and the model is trained with early stopping if the model does not improve in test loss within three epochs. In practice, the model took, on average 25 minutes to train on a 56-Core 16 GB Tesla P100 GPU.

C.2. Raw Data Results

This subsection of the appendix reports the raw values for DIR and the four properties for each datum, separated by algorithm-dataset interaction. Table C.5 gives the raw sum of squared errors for the individual property models and the entire models used to craft the hypothesis tests.

Part	Layer	Details
Encoder	Conv2d	In: 3, Out: 16, Kernel: 3x3, Stride: 2, Padding: 1, Bias: False
	ReLU	In-place: True
	Conv2d	In: 16, Out: 32, Kernel: 3x3, Stride: 2, Padding: 1, Bias: False
	BatchNorm2d	Num Features: 32
	ReLU	In-place: True
	Conv2d	In: 32, Out: 64, Kernel: 3x3, Stride: 2, Padding: 0, Bias: False
	ReLU	In-place: True
	Flatten	Start Dim: 1
	Linear	In: 38016, Out: 128, Bias: False
	ReLU	In-place: True
	Linear	In: 128, Out: Encoded Space Dim, Bias: False
Decoder	Linear	In: Encoded Space Dim, Out: 128
	ReLU	In-place: True
	Linear	In: 128, Out: 38016
	ReLU	In-place: True
	Unflatten	Dim: 1, Unflattened Size: (64, 22, 27)
	ConvTranspose2d	In: 64, Out: 32, Kernel: 3x3, Stride: 2, Output Padding: 0
	BatchNorm2d	Num Features: 32
	ReLU	In-place: True
	ConvTranspose2d	In: 32, Out: 16, Kernel: 3x3, Stride: 2, Padding: 1
	BatchNorm2d	Num Features: 16
	ReLU	In-place: True
	ConvTranspose2d	In: 16, Out: 3, Kernel: 3x3, Stride: 2, Padding: 1, Output Padding: 1
	Sigmoid	

TABLE C.1. Unfairness and property values for CelebA Attributes via Autoencoder

Attribute	Unfairness (DIR)	Reconstruction Ratio	SSB	SFV	Label Noise
5_o_Clock_Shadow	1.118	1.183	0.8904	0.2252	0.4869
Arched_Eyebrows	1.124	1.0033	0.7252	0.2287	0.4869
Attractive	1.075	1.1356	0.5122	0.2339	0.486
Bags_Under_Eyes	1.308	1.077	0.799	0.2259	0.6119
Bald	1.164	1.1017	0.9766	0.2233	0.5019
Bangs	1.059	1.1121	0.8518	0.2414	0.5687
Big_Lips	1.219	1.1	0.7534	0.2262	0.2721
Big_Nose	1.457	1.1	0.7684	0.2247	0.4415
Black_Hair	1.007	1.15	0.7568	0.2248	0.5283
Blond_Hair	1.042	1.14	0.854	0.2276	0.4273
Blurry	1.128	1.19	0.946	0.2406	1.0181
Brown_Hair	1.080	1.11	0.7962	0.224	0.6281
Bushy_Eyebrows	1.088	1.17	0.859	0.2248	0.5107
Chubby	2.992	1.23	0.942	0.2236	0.6076
Double_Chin	1.413	1.26	0.9578	0.2244	0.709
Eyeglasses	1.600	1.35	0.937	0.2244	0.585
Goatee	1.479	1.27	0.9368	0.2346	0.4938
Gray_Hair	1.053	1.19	0.952	0.2247	0.5767
Heavy_Makeup	1.100	1	0.6148	0.229	0.3698
High_Cheekbones	1.547	1.06	0.5536	0.235	0.6822
Male	1.117	1.01	0.5834	0.2236	0.0211
Mouth_Slightly_Open	1.058	1.08	0.5222	0.2262	0.7859
Mustache	1.280	1.3	0.9616	0.2409	0.5055
Narrow_Eyes	1.017	1.18	0.8808	0.2252	0.7622
No_Beard	3.201	1.43	0.8322	0.231	0.355
Oval_Face	1.672	1.07	0.7296	0.2285	0.6119
Pale_Skin	1.491	1.17	0.9586	0.2367	0.8438
Pointy_Nose	1.301	1.08	0.732	0.2272	0.5454
Receding_Hairline	1.576	1.15	0.9228	0.2248	0.6595
Rosy_Cheeks	1.035	1.19	0.9382	0.2248	0.6718
Sideburns	1.553	1.27	0.9396	0.2286	0.5241
Smiling	1.317	1.07	0.5188	0.239	0.7449
Straight_Hair	1.118	1.17	0.7874	0.2268	0.6559
Wavy_Hair	1.488	1.02	0.69	0.2239	0.5728
Wearing_Earrings	1.052	1.11	0.8064	0.2269	0.6107
Wearing_Hat	1.645	1.32	0.9512	0.2269	0.7502
Wearing_Lipstick	1.213	1.07	0.5288	0.2286	0.2678
Wearing_Necklace	1.349	1.18	0.8686	0.2246	0.6887
Wearing_Necktie	1.077	1.18	0.9244	0.2260	0.7288
Young	1.997	1.36	0.7826	0.2250	0.164

TABLE C.2. Unfairness and property values for LFW Attributes via Autoencoder

	Unfairness (DIR)	Reconstruction Ratio	SSB	SFV	Label Noise
Male	1.12200367380267	1.00535333156585	0.774632884425169	0.2031201482	0.07679999999999998
Asian	1.053645403248	1.1167961359024	0.92322909533592	0.2021178782	
White	1.1264462529671	1.01018273830413	0.747926652971163	0.2088530302	
Black	1.12365634206545	1.18320667743682	0.957391767480788	0.2025963485	0.08120000000000005
Baby	1.06544960186443	1.11203300952911	0.836643079966522	0.2036614358	0.09550000000000003
Child	1.12626262626262	1.12434077262878	0.898196758730883	0.2015907168	0.10470000000000002
Youth	1.1684570024365	1.09732460975646	0.786274062238453	0.2201078296	0.13770000000000004
Middle Aged	1.05672615298764	1.10370337963104	0.866316670470973	0.2059026539	0.0645
Senior	1.77747312898089	1.1931574344635	0.957467853610286	0.2064542949	0.16779999999999995
Black Hair	1.12004451070385	1.07356524467468	0.63349311420528	0.2042071939	
Blond Hair	1.05108769459044	1.13573598861694	0.891653351594004	0.2307599187	
Brown Hair	1.03576168696236	1.01701772212982	0.7891653351594	0.2089367867	
Bald	1.00087648056866	1.13213109970092	0.824621471505744	0.2131045997	0.11350000000000005
No Eyewear	1.08863610960647	1.14704251289367	0.985087118618275	0.2012821913	0.06899999999999995
Eyeglasses	1.06348181302805	1.1471596956253	0.877729589895762	0.2019755244	0.19679999999999997
Sunglasses	1.05997138025237	1.06211602687835	0.586700144563646	0.2035428464	0.20889999999999997
Mustache	1.0421456164088	1.03791272640228	0.581298029369246	0.2034806907	0.21950000000000003
Smiling	1.14102186869087	1.07792913913726	0.641101727155139	0.2054580331	0.2974
Frowning	1.16748745804309	1.10730016231536	0.843262573232899	0.2032266498	0.07879999999999998
Chubby	1.08701997540087	1.08353006839752	0.683557787415354	0.2049415469	0.10560000000000003
Blurry	1.04091852227881	1.10518634319305	0.811154226584493	0.2110888839	0.27580000000000005
Harsh Lighting	1.05681504499685	1.02375900745391	0.695579395876131	0.2157218277	0.30279999999999996
Soft Lighting	1.05644459380154	1.03066658973693	0.598417408506429	0.2086786151	0.15849999999999997
Outdoor	1.06132796694575	1.07666659355163	0.566613406376017	0.221262145	0.22760000000000002
Curly Hair	1.01377517221455	1.06283998489379	0.62375408962946	0.2088014245	0.14180000000000004
Wavy Hair	1.18200199173129	1.03124058246612	0.581830632275736	0.2119037926	0.04500000000000004
Straight Hair	1.25206733987405	1.16164600849151	0.835806132542037	0.2045027018	0.25670000000000004
Receding Hairline	1.132162388614	1.08063757419586	0.690329452940728	0.2026151061	0.31120000000000003
Bangs	1.16524283964575	1.00697135925292	0.672981815415049	0.2061040878	0.33009999999999995
Sideburns	1.12084015275504	1.16936266422271	0.939739785437114	0.2015084624	0.22319999999999995
Fully Visible Forehead	1.21900390887339	1.1811419725418	0.858784143650612	0.2038781226	0.2298
Partially Visible Forehead	1.05020804838356	1.04769682884216	0.536483299094575	0.2050496221	0.15200000000000002
Obstructed Forehead	1.197095435684	1.09454452991485	0.73674199193487	0.2027293146	0.11260000000000003
Bushy Eyebrows	1.21132478772795	1.02557587623596	0.645286464277562	0.2025717795	0.0998
Arched Eyebrows	1.11604546137808	1.02139496803283	0.862360191737046	0.2056749165	0.15269999999999995
Narrow Eyes	1.01965937186759	1.00794005393981	0.69078596971772	0.2016550004	0.19099999999999995
Eyes Open	1.21624935631726	1.01386857032775	0.698166324279083	0.2054687798	0.2893
Big Nose	1.02608985048702	1.0604817867279	0.634406147759263	0.205928582	0.06599999999999995
Pointy Nose	1.19868957288718	1.07568454742431	0.622764969945978	0.2045042574	0.046699999999999964
Big Lips	1.06428433432607	1.06256353855133	0.663166704709731	0.2033233523	0.08440000000000003
Mouth Closed	1.01175554129597	1.12742841243743	0.904511907479266	0.2021872401	0.32189999999999996
Mouth Slightly Open	1.06630991503093	1.04525172710418	0.571102488016434	0.2044097185	0.07479999999999998
Mouth Wide Open	1.01637465524165	1.01459431648254	0.714981358898272	0.2013282001	0.18810000000000004
Teeth Not Visible	1.09729244959597	1.1062124967575	0.759796089172943	0.2011253536	0.27669999999999995
No Beard	1.0605139319402	1.01765537261962	0.869284029521418	0.2096437275	0.2319
Goatee	1.11854311102431	1.08315765857696	0.639351746176672	0.2126889467	0.04530000000000001
Round Jaw	1.12212437767378	1.15832161903381	0.860229780111085	0.2023314357	0.050899999999999945
Double Chin	1.01444585996835	1.04857730865478	0.519135661568896	0.2015054762	0.1965
Wearing Hat	1.31578440808469	1.13376498222351	0.950467929696416	0.2047562778	0.08360000000000001
Oval Face	1.227980920874	1.10575580596923	0.920718253062466	0.2021733284	0.12819999999999998
Square Face	1.08180300500834	1.03950214385986	0.957087422962793	0.2091827631	0.08360000000000001
Round Face	1.00385912356425	1.03282678127288	0.504983641482157	0.2034206629	0.2126
Color Photo	1.03475440467016	1.07052874565124	0.664764513429201	0.2038946807	0.10570000000000002
Posed Photo	1.05681639747742	1.10381340980529	0.848740774556798	0.2014933527	0.15300000000000002
Attractive Man	1.15967929714224	1.09643280506134	0.977098075020923	0.2019033909	0.35509999999999997
Attractive Woman	1.08906867243748	1.10497522354125	0.84105607547744	0.227733314	0.13529999999999998
Indian	1.01517435331474	1.03677427768707	0.588830556189606	0.2017122924	0.14029999999999998
Gray Hair	1.0282016857369	1.10868871212005	0.882523016054173	0.2012593031	0.18779999999999997
Bags Under Eyes	1.00131664057342	1.11418402194976	0.805143422354104	0.2055422544	0.13219999999999998
Heavy Makeup	1.14755164575804	1.11894488334655	0.879707829262725	0.2015730679	0.21609999999999996
Rosy Cheeks	1.02025763283369	1.05831480026245	0.507037966978619	0.2184995234	0.08520000000000005
Shiny Skin	1.09951096814278	1.06581234931945	0.591797915240051	0.2018399835	0.10729999999999995
Pale Skin	1.06768325049461	1.05000007152557	0.534428973598113	0.2011277676	0.1421
5 o' Clock Shadow	1.16855307810665	1.09448540210723	0.84432777904588	0.2016790688	
Strong Nose-Mouth Lines	1.03358017791439	1.10516810417175	0.866697101118466	0.2044023335	
Wearing Lipstick	1.08893014058315	1.07446813583374	0.656471125313855	0.2032339275	
Flushed Face	1.0144694850683	1.01248931884765	0.655482005630373	0.2137254417	
High Cheekbones	1.00916172995591	1.12738478183746	0.860229780111085	0.2017118096	
Brown Eyes	1.08648174717041	1.01540994644165	0.636384387126226	0.2055488884	
Wearing Earrings	1.10247725115406	1.09833109378814	0.79563265616678	0.2028558612	

TABLE C.3. Unfairness and property values for CelebA Attributes via Deep SVDD

	SVDD	Reconstruction	SSB	Spurious Feature Variance	Label Noise
5_o_Clock_Shadow	1.68123553498308	1.46047670114505	0.8904	0.1409505606	0.4869
Arched_Eyebrows	1.25764192139738	1.29294249928091	0.7252	0.1452494413	0.4869
Attractive	1.09368792760979	1.05381571022971	0.5122	0.1520317346	0.486
Bags_Under_Eyes	1.06134410518395	1.12471149407601	0.798999999999999	0.1400723457	0.6119
Bald	2.352	1.02880658436214	0.9766	0.1451713741	0.5019
Bangs	1.39449541284403	1.32236633976589	0.8518	0.1576949656	0.5687
Big_Lips	1.70156624102154	1.08017998183669	0.7534	0.1442556977	0.2721
Big_Nose	1.30569948186528	1.16960464068483	0.7684	0.1389202923	0.4415
Black_Hair	1.00635593220339	1.10986682808716	0.7568	0.1428498179	0.5283
Blond_Hair	1.03992089562244	1.29937377627469	0.854	0.1420869976	0.4273
Blurry	1.35800508259212	1.12928843710292	0.946	0.1826313585	1.0181
Brown_Hair	1.07992104600792	1.00426740416926	0.7962	0.1399643421	0.6281
Bushy_Eyebrows	1.26066424494032	1.1293009118541	0.859	0.1396305859	0.5106999999999999
Chubby	1.15950659293917	1.0393457117595	0.942	0.143850103	0.6075999999999999
Double_Chin	1.46185598532334	1.24815246204514	0.9578	0.1393095106	0.7090000000000001
Eyeglasses	1.4847619047619	1.13053239255933	0.937	0.1726125926	0.585
Goatee	1.30087633885102	1.29235531479741	0.9368	0.148198694	0.4938
Gray_Hair	2.44949494949494	1.63565217391304	0.952	0.1400457323	0.5767
Heavy_Makeup	1.37794331165961	1.21201795786807	0.6148	0.1471818388	0.3698
High_Cheekbones	1.41521739130434	1.07322226737098	0.5536	0.148946777	0.6821999999999999
Male	1.16378620579292	1.12330668559143	0.583399999999999	0.1411117315	0.021100000000000008
Mouth_Slightly_Open	1.47328992862486	1.01889931435045	0.5222	0.1415492892	0.7859
Mustache	1.28	1.04602510460251	0.9616	0.1618342251	0.5055000000000001
Narrow_Eyes	1.3557779799818	1.08768131630222	0.8808	0.1405434906	0.7622
No_Beard	1.26765068774848	1.32170279829207	0.8322	0.1428951621	0.355
Oval_Face	1.00961538461538	1.05142857142857	0.7296	0.1448870301	0.6119
Pale_Skin	1.49135109864422	1.06838387528924	0.9586	0.1604245007	0.8438
Pointy_Nose	1.28422782037239	1.26457127210139	0.732	0.1431550533	0.5454
Receding_Hairline	1.10142050741269	1.33176813471502	0.9228	0.1404222101	0.6595
Rosy_Cheeks	1.153123680878	1.25196285352469	0.9382	0.1406327337	0.6718
Sideburns	1.36525725929699	1.50509087726463	0.9396	0.1386207491	0.5241
Smiling	1.11647331786542	1.01603413341645	0.518799999999999	0.1510140896	0.7449
Straight_Hair	1.13916759320035	1.16279926135717	0.7874	0.1428056359	0.6558999999999999
Wavy_Hair	1.6726155889433	1.30170504067402	0.69	0.1401683241	0.5728
Wearing_Earrings	1.08250497017892	1.01847107438016	0.8064	0.1419264823	0.6107
Wearing_Hat	5.03622577927548	1.54812552653748	0.9512	0.2158842981	0.7502
Wearing_Lipstick	1.26436951774677	1.16687742370595	0.528799999999999	0.1471352577	0.26780000000000004
Wearing_Necklace	1.00260846420015	1.07914052831476	0.8686	0.1395401657	0.6887
Wearing_Necktie	1.52579365079365	1.36231575963718	0.9244	0.1407860667	0.7288
Young	1.0892026578073	1.17668546526531	0.7826	0.1402778327	0.16400000000000003

TABLE C.4. Unfairness and property values for LFW Attributes via Deep SVDD

	DIR	Incompressibility	SSB	SFV	Label Noise
Male	1.17931562745317	1.0924447774887	0.774632884425169	0.1429237619	0.07679999999999998
Asian	1.23055692048871	1.01016497611999	0.92322909533592	0.1415031001	
White	1.00406917599186	1.08468961715698	0.747926652971163	0.1511053567	
Black	1.34819532908704	1.03384220600128	0.957391767480788	0.1421764963	0.08120000000000005
Baby	1.06271364829537	1.03743159770965	0.836643079966522	0.143683949	0.09550000000000003
Child	1.13670569529881	1.03432464599609	0.898196758730883	0.140765438	0.10470000000000002
Youth	1.05082822021653	1.03086674213409	0.786274062238453	0.1678981342	0.13770000000000004
Middle Aged	1.10867550207333	1.02771830558776	0.866316670470973	0.1468662707	0.0645
Senior	1.00186866902908	1.04160547256469	0.957467853610286	0.1476565742	0.16779999999999995
Black Hair	1.02750194844192	1.04895257949829	0.63349311420528	0.1444525348	
Blond Hair	1.08838038386602	1.01417303085327	0.891653351594004	0.184528688	
Brown Hair	1.03209559606518	1.09958708286285	0.7891653351594	0.1512480983	
Bald	1.00790551940226	1.00073754787445	0.824621471505744	0.1573695429	0.11350000000000005
No Eyewear	1.40606623336428	1.00252616405487	0.985087118618275	0.1403350747	0.06899999999999995
Eyeglasses	1.43913177607322	1.00310981273651	0.877729589895762	0.1413070618	0.19679999999999997
Sunglasses	1.12142575468585	1.05022633075714	0.586700144563646	0.1435135175	0.20889999999999997
Mustache	1.11839255634876	1.07019913196563	0.581298029369246	0.1434256518	0.21950000000000003
Smiling	1.07582623948232	1.03398072719573	0.641101727155139	0.1462316354	0.2974
Frowning	1.38021050679278	1.04254591464996	0.843262573232899	0.14306304	0.07879999999999998
Chubby	1.37894686222649	1.04961144924163	0.683557787415354	0.1454682116	0.10560000000000003
Blurry	1.07484216395665	1.01648831367492	0.811154226584493	0.1543799572	0.27580000000000005
Harsh Lighting	1.76541734255385	1.06726253032684	0.695579395876131	0.1612156139	0.30279999999999996
Soft Lighting	1.15505859850802	1.0463809967041		0.1456943556	0.1642
Outdoor	1.20495433082845	1.06911957263946	0.598417408506429	0.1508525053	0.15849999999999997
Curly Hair	1.13923719958202	1.05589497089385	0.566613406376017	0.1696171921	0.22760000000000002
Wavy Hair	1.06940992787003	1.04987812042236	0.62375408962946	0.1510261253	0.14180000000000004
Straight Hair	1.04934265833276	1.07179963588714	0.581830632275736	0.1555814983	0.04500000000000004
Receding Hairline	1.17698276832539	1.00636541843414	0.835806132542037	0.1448722679	0.25670000000000004
Bangs	1.09157918248827	1.06093919277191	0.690329452940728	0.1422119786	0.31120000000000003
Sideburns	1.15947653456037	1.08820021152496	0.672981815415049	0.1471594972	0.33009999999999995
Fully Visible Forehead	1.55668147556531	1.00061905384063	0.939739785437114	0.1406534992	0.22319999999999995
Partially Visible Forehead	1.25747607655502	1.03077602386474	0.858784143650612	0.1439849571	0.2298
Obstructed Forehead	1.11325281649095	1.05480468273162	0.536483299094575	0.1456306697	0.15200000000000002
Bushy Eyebrows	1.00786702803827	1.03234314918518	0.73674199193487	0.1423631794	0.11260000000000003
Arched Eyebrows	1.06208761023718	1.07342624664306	0.645286464277562	0.1421420989	0.0998
Narrow Eyes	1.08786442753544	1.09905493259429	0.862360191737046	0.1465277227	0.15269999999999995
Eyes Open	1.223370100546	1.07901871204376	0.69078596971772	0.1408611888	0.19099999999999995
Big Nose	1.03111518672274	1.0845707654953	0.698166324279083	0.1462382611	0.2893
Pointy Nose	1.11446611115883	1.05301141738891	0.634406147759263	0.1469018736	0.06599999999999995
Big Lips	1.14029929024963	1.05008065700531	0.622764969945978	0.1448668399	0.046699999999999964
Mouth Closed	1.0816224959562	1.06419742107391	0.663166704709731	0.1431995614	0.08440000000000003
Mouth Slightly Open	1.01337628971086	1.03440833091735	0.904511907479266	0.1416061479	0.32189999999999996
Mouth Wide Open	1.03990024937655	1.06561398506164	0.571102488016434	0.1447347991	0.07479999999999998
Teeth Not Visible	1.04770316767762	1.07595670223236	0.714981358898272	0.1403963187	0.18810000000000004
No Beard	1.0876431987543	1.03276085853576	0.759796089172943	0.1401160741	0.27669999999999995
Goatee	1.19802672343941	1.09399461746215	0.869284029521418	0.1522871416	0.2319
Round Jaw	1.06897059287373	1.04447555541992	0.639351746176672	0.156724269	0.04530000000000001
Double Chin	1.02390223246378	1.00809562206268	0.860229780111085	0.141805179	0.050899999999999945
Wearing Hat	1.08353184055899	1.06220078468322	0.519135661568896	0.1406492755	0.1965
Oval Face	1.12880495352612	1.02028930187225	0.950467929696416	0.1452499895	0.08360000000000001
Square Face	1.03973957569458	1.0401998758316	0.920718253062466	0.141584407	0.12819999999999998
Round Face	1.1325489572568	1.10276663303375	0.957087422962793	0.1514307107	0.08360000000000001
Color Photo	1.13798432728612	1.07599568367004	0.504983641482157	0.143338242	0.2126
Posed Photo	1.0394726007875	1.0795783996582	0.664764513429201	0.1440141143	0.10570000000000002
Attractive Man	1.08110687391574	1.03554499149322	0.848740774556798	0.1406362299	0.15300000000000002
Attractive Woman	1.51685778921912	1.05630433559417	0.977098075020923	0.1412037472	0.35509999999999997
Indian	1.04062717938913	1.05820059776306	0.84105607547744	0.1797280974	0.13529999999999998
Gray Hair	1.1973171397336	1.06187999248504	0.588830556189606	0.1409442876	0.14029999999999998
Bags Under Eyes	1.45747314192372	1.02336192131042	0.882523016054173	0.140306542	0.18779999999999997
Heavy Makeup	1.00048536793256	1.02952170372009	0.805143422354104	0.1463585953	0.13219999999999998
Rosy Cheeks	1.16560850348722	1.03843355178833	0.879707829262725	0.1407405867	0.21609999999999996
Shiny Skin	1.26315502068762	1.05610179901123	0.507037966978619	0.1653680619	0.08520000000000005
Pale Skin	1.18351263647553	1.0652779340744	0.591797915240051	0.1411130869	0.10729999999999995
5 o' Clock Shadow	1.17272883141058	1.05673563480377	0.534428973598113	0.1401202399	0.1421
Strong Nose-Mouth Lines	1.22455279703978	1.03042745590209	0.84432777904588	0.1408965065	
Wearing Lipstick	1.19464912714504	1.04016602039337	0.866697101118466	0.1447695088	
Flushed Face	1.09233277829563	1.04554724693298	0.656471125313855	0.1430755171	
High Cheekbones	1.11059337257828	1.08112835884094	0.655482005630373	0.1582355048	
Brown Eyes	1.17224142515288	1.01272892951965	0.860229780111085	0.1409409852	
Wearing Earrings	1.16634746922024	1.0849984884262	0.636384387126226	0.1463791189	

TABLE C.5. Squared Error (SE) For Various Properties & The Proposed Whole Model

	Incompressibility	SSB	SFV	Label Noise	Whole Model
5_o_Clock_Shadow	0.0684682	0.0459237	0.0002894	3e-7	3e-7
Arched_Eyebrows	0.0126105	0.0538039	0.0131643	0.0106393	0.0106393
Attractive	0.0004621	0.0338707	0.0483136	0.0275897	0.0004621
Bags_Under_Eyes	0.0202342	0.0623039	0.0050525	0.0093071	0.0050525
Bald	0.4319569	0.00024	0.0943209	0.0748498	0.00024
Bangs	0.0006496	0.1283069	0.0774581	0.0298166	0.0006496
Big_Lips	0.1088419	0.0091254	0.0032081	0.0052196	0.0032081
Big_Nose	0.2114748	0.0001715	0.0252706	0.020549	0.0001715
Black_Hair	0.0142123	0.0593484	0.0061956	0.0106132	0.0061956
Blond_Hair	0.0100063	0.0946047	0.0130189	0.0110102	0.0100063
Blurry	0.5419093	0.0074364	0.0490523	0.0729299	0.0074364
Brown_Hair	0.0001064	0.1154422	0.0218438	0.0360707	0.0001064
Bushy_Eyebrows	0.0213848	0.0773384	0.0035219	0.006768	0.0035219
Chubby	0.0013206	0.1581314	0.015918	0.0287418	0.0013206
Double_Chin	0.0035046	0.1521108	0.0137349	0.0272636	0.0035046
Eyeglasses	0.0113092	0.120019	0.0071329	0.0140104	0.0071329
Goatee	0.1604753	0.0201623	0.0013985	0.0092697	0.0013985
Gray_Hair	0.0002775	0.1933109	0.0296317	0.0411467	0.0002775
Heavy_Makeup	0.0000469	0.0613383	0.0363665	0.0266154	0.0000469
High_Cheekbones	0.0930655	0.0000754	0.0010459	0.0000749	0.0000749
Male	0.0336658	0.0118035	0.0005703	0.0000012	0.0000012
Mouth_Slightly_Open	0.0443864	0.0033491	0.0006975	0.0047076	0.0006975
Mustache	0.282736	0.0040766	0.0060553	0.0346049	0.0040766
Narrow_Eyes	0.2105005	0.005241	0.0237357	0.0113154	0.005241
No_Beard	0.6583177	0.03911	0.1239485	0.1577867	0.03911
Oval_Face	0.3213105	0.0066273	0.0411204	0.0393565	0.0066273
Pale_Skin	0.2731508	0.0047057	0.0111297	0.0200174	0.0047057
Pointy_Nose	0.0001962	0.0922673	0.0293014	0.0316888	0.0001962
Receding_Hairline	0.5347036	0.0090082	0.1198915	0.0943515	0.0090082
Rosy_Cheeks	0.0309238	0.0892779	0.0015001	0.0063849	0.0015001
Sideburns	0.1497514	0.023381	0.005897	0.0069555	0.005897
Smiling	0.1429648	0.0036202	0.0001827	0.0027342	0.0001827
Straight_Hair	0.2121866	0.0005044	0.0203373	0.0143183	0.0005044
Wavy_Hair	0.0181157	0.0392565	0.0036124	0.0094807	0.0036124
Wearing_Earrings	0.0000825	0.1285602	0.0341021	0.0405731	0.0000825
Wearing_Hat	0.0069229	0.1368304	0.0145395	0.0234406	0.0069229
Wearing_Lipstick	0.0278786	0.0084025	0.0057513	0.0016339	0.0016339
Wearing_Necklace	0.0691438	0.0408226	0.0005308	0.0004549	0.0004549
Wearing_Necktie	0.1449726	0.0222941	0.0086847	0.00313	0.00313
Young	0.1940491	0.0011516	0.0204862	0.0260069	0.0011516

Bibliography

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

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Playing atari with deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] Jason Smucny, Ge Shi, and Ian Davidson. Deep learning in neuroimaging: Overcoming challenges with emerging approaches. *Frontiers in Psychiatry*, 13:912600, 2022.

[4] Jason Smucny, Ge Shi, Tyler A Lesh, Cameron S Carter, and Ian Davidson. Data augmentation with mixup: Enhancing performance of a functional neuroimaging-based prognostic deep learning classifier in recent onset psychosis. *NeuroImage: Clinical*, 36:103214, 2022.

[5] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. *IEEE Intelligent Transportation Systems Magazine*, 3(4):8–19, 2011.

[6] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2016.

[7] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[8] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, Vaishaal Shankar, Shiori Zhang, Ari S Morcos, and Yoshua Bengio. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

[9] Robert Geirhos, Jannis-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.

[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[11] Brian Christian. *The Alignment Problem: Machine Learning and Human Values*. W. W. Norton & Company, 2020.

[12] Christoph Molnar. *Interpretable Machine Learning*. self-published, 2 edition, 2022.

[13] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, November 2021.

[14] 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. ACM, 2016.

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626. IEEE, 2017.

[16] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree, 2017.

[17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[18] Shuyang Liu, Zixuan Chen, Ge Shi, Ji Wang, Changjie Fan, Yu Xiong, Runze Wu Yujing Hu, Ze Ji, and Yang Gao. A new baseline assumption of integated gradients based on shaply value, 2024.

[19] Michal K. Grzeszczyk, Tomasz Trzciński, and Arkadiusz Sitek. Miss: Multiclass interpretable scoring systems, 2024.

[20] Luca Oneto and Silvia Chiappa. *Fairness in Machine Learning*, page 155–196. Springer International Publishing, 2020.

[21] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7), apr 2024.

[22] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, pages 1–7. ACM, May 2018.

[23] Simon Caton and Christian Haas. Fairness in machine learning: A survey, 2020.

[24] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, 2019.

[25] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in ai. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 279–288, New York, NY, USA, 2019. Association for Computing Machinery.

[26] Fair credit reporting act, 1970. 15 U.S.C. §§ 1681-1681x.

[27] European Parliament and of the Council. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. Retrieved from https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[29] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[30] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

[31] Usman Gohar and Lu Cheng. A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, August 2023.

[32] Hongjing Zhang and Ian Davidson. Towards fair deep anomaly detection, 2020.

[33] Kimberlé Williams Crenshaw. Mapping the margins: Intersectionality, identity politics, and violence against women of color. *Stanford Law Review*, 43(6):1241–1299, 1991.

[34] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, 2018.

[35] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018.

[36] Michael Wooldridge, Jennifer Dy, and Sriraam Natarajan, editors. *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, Fourteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, Washington, DC, USA, February 20–27 2024. Sponsored by the Association for the Advancement of Artificial Intelligence.

[37] Gregory L. Murphy. *The Big Book of Concepts*. The MIT Press, January 2004.

[38] Shashi Shekhar, Vagelis Papalexakis, Jing Gao, Zhe Jiang, Shashi Shekhar Riondato, Matteo, Vagelis Papalexakis, Jing Gao, Zhe Jiang, and Matteo Riondato, editors. *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2024.

[39] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *WIREs Data Mining and Knowledge Discovery*, 12(3), 2022.

[40] Michael Livanos, Ian Davidson, and Stephen Wong. Cooperative knowledge distillation: A learner agnostic approach. *arXiv preprint arXiv:2402.05942*, 2024.

[41] Geoffrey et Al Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[42] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[43] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[44] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6), Mar 2021.

[45] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[46] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer, 2019.

[47] Jangho Kim and SeongUk Park. Paraphrasing complex network: Network compression via factor transfer. *arXiv preprint arXiv:1802.04977*, 2020.

[48] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), Jul. 2019.

[49] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[51] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

[52] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[53] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

[54] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[55] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3430–3437, 2020.

[56] Haoran Zhang, Zhenzhen Hu, Wei Qin, Mingliang Xu, and Meng Wang. Adversarial co-distillation learning for image recognition. *Pattern Recognition*, 111:107659, 2021.

[57] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[58] Zhilu Zhang and Mert R Sabuncu. Self-distillation as instance-specific label smoothing. *arXiv preprint arXiv:2006.05065*, 2020.

[59] Mark T Keane and Barry Smyth. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In *International Conference on Case-Based Reasoning*. Springer, 2020.

[60] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95*, volume 4, pages 1942–1948 vol.4, 1995.

[61] Paul D. Allison. *Missing Data*. Sage Publications, 2001.

[62] Piotr S. Gromski, Yun Xu, Helen L. Kotze, Elon Correa, David I. Ellis, Emily Grace Armitage, Michael L. Turner, and Royston Goodacre. Influence of missing values substitutes on multivariate analysis of metabolomics data. *Metabolites*, 4(2):433–452, 2014.

[63] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[64] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2016.

[65] Insaf Ashrapov. Tabular gans for uneven distribution. *arXiv preprint arXiv:2010.00638*, 2020.

[66] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017.

[67] Austin Reese. Used cars dataset vehicles listings from craigslist.org, 2021. https://www.kaggle.com/austinreese/craigslist-carstrucks-data.

[68] Doaa Alsenani. Us cars dataset: Online car auction in north american, 2020. Retrieved from https://www.kaggle.com/doaaalsenani/usa-cers-dataset.

[69] Ananay Mital. Us used cars dataset, 2020. https://www.kaggle.com/ananaymital/us-used-cars-dataset.

[70] Andras Janosi M.D., William Steinbrunn M.D., Matthias Pfisterer M.D., and Robert Detrano M.D. Ph.D. Heart data set, 1988.

[71] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. http://archive.ics.uci.edu/ml.

[72] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[73] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[74] Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

[75] Xianzhi Du, Wei-Chih Hung, and Tsung-Yi Lin. Optimizing anchor-based detectors for autonomous driving scenes, 2022.

[76] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[77] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[78] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *ACM SIGMOD International Conference On Management of Data*, 2000.

[79] Shebuti Rayana. Odds (outlier detection data sets) library, 2016.

[80] Joe Sipple and A. Youssef. A general-purpose method for applying explainable ai for anomaly detection, 2022.

[81] Jannis Kauffmann, Lukas Ruff, Grégoire Montavon, and Klaus-Robert Müller. The clever hans effect in anomaly detection, 2020.

[82] Daniel Sulem, Michele Donini, Muhammad Bilal Zafar, François-Xavier Aubet, Jan Gasthaus, Tim Januschowski, and Cedric Archambeau. Diverse counterfactual explanations for anomaly detection in time series, 2022.

[83] Jannis Kauffmann, Klaus-Robert Müller, and Grégoire Montavon. Towards explaining anomalies: a deep taylor decomposition of one-class models, 2020.

[84] Audrey Der, Chin-Chia Michael Yeh, Yan Zheng, Junpeng Wang, Zhongfang Zhuang, Liang Wang, Wei Zhang, and Eamonn J. Keogh. Pupae: Intuitive and actionable explanations for time series anomalies, 2024.

[85] Liat Friedman Antwarg, Ronnie Miller, Bracha Shapira, and Lior Rokach. Explaining anomalies detected by autoencoders using shapley additive explanations. *Expert Systems with Applications*, 186:115736, 08 2021.

[86] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[87] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–54, 2023.

[88] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 2019.

[89] Guansong et. al Pang. Explainable deep few-shot anomaly detection with deviation networks, 2021.

[90] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. Robust and explainable autoencoders for unsupervised time series outlier detection—extended version, 2022.

[91] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

[92] Roman Zwicky. A brief introduction to dispersion relations and analyticity. *arXiv preprint arXiv:1610.06090*, 2016.

[93] SP Lloyd. Least square quantization in pcm. *IEEE Trans. Inform. Theor.(1957/1982)*, 18:5, 1957.

[94] Igor Kononenko, Bojan Cestnik. Lymphography data set, 1988.

[95] Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian, and Computer Sciences Dept. Breast cancer wisconsin (diagnostic) data set, 1995.

[96] Marques de SA, J.P Bernardes, and J. Ayres de Campos D. Cardiography data set, 2010.

[97] B. German. Glass identification data set, 1987.

[98] David J. Slate. Letter recognition data set, 1991.

[99] K. Woods, C. Doss, K. Bowyer, J. Solka, and C. Priebe. Mammography data set, 2014.

[100] Ajay Jain David Chapman. Musk v2 dataset, 1994.

[101] NASA Ashwin Srinivasan. Statlog (landsat satellite) data set, 1993.

[102] Jason Catlett. Statlog (shuttle) data set, 1996.

[103] Barbora Micenkova, Brian McWilliams, and Ira Assent. Speech data set, 2014. Brno University of Technology, Czech Republic.

[104] Marek Sikora. Seismic bumps data set, 2013.

[105] M. Forina. Wine data set, 1991.

[106] Volker Lohweg. Banknote authentication data set, 2013.

[107] Murat Koklu Ilkay Cinar. Rice (cammeo and osmancik) data set, 2019.

[108] Rajen Bhatt. Wireless indoor localization data set, 2017.

[109] Rifkie Primartha and Bayu Adhi Tama. Anomaly detection using random forest: A performance revisited. In *2017 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6, 2017.

[110] Thorben Finke et al. Autoencoders for unsupervised anomaly detection in high energy physics. *Journal of High Energy Physics*, 2021(6), jun 2021.

[111] Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. A survey on gans for anomaly detection, 2019.

[112] I. Davidson, M. Livanos, A. Gourru, P. Walker, J. Velcin, and S. S. Ravi. Explainable clustering via exemplars: Complexity and efficient approximation algorithms. ArXiv: 2209.09670, Primary Class: cs.AI, 2022.

[113] D. Gunning and D. W Aha. DARPA's explainable artificial intelligence program. *AI Magazine*, 40(2):44–58, 2019.

[114] 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, 2016.

[115] S. Dasgupta, N. Frost, M. Moshkovitz, and C. Rashtchian. ExKMC: Expanding explainable k-means clustering. *ICML*, 2020.

[116] R. S Michalski and R. E. Stepp. Learning from observation: Conceptual clustering. In *Machine Learning*, pages 331–363. Springer, 1983.

[117] S. Saisubramanian, S. Galhotra, and S. Zilberstein. Balancing the tradeoff between clustering value and interpretability. In *Proc. AIES Conference*, pages 351–357, 2020.

[118] I. Davidson, A. Gourru, and S. S. Ravi. The cluster description problem-complexity results, formulations and approximations. In *Advances in Neural Information Processing Systems*, pages 6190–6200, 2018.

[119] M. Walsh, B. Möbius, T. Wade, and H. Schütze. Multilevel exemplar theory. *Cognitive science*, 34(4):537–582, 2010.

[120] Y. Wang and L. Chen. K-MEAP: Generating specified K clusters with multiple exemplars by efficient affinity propagation. In *Proc. IEEE International Conference on Data Mining (ICDM)*, pages 1091–1096, 2014.

[121] F. G. Ashby and W. T. Maddox. Human category learning. *Annu. Rev. Psychol.*, 56:149–178, 2005.

[122] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[123] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., San Francisco, 1979.

[124] V. V. Vazirani. *Approximation Algorithms*. Springer, New York, NY, 2001.

[125] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.

[126] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70:39–45, 1999.

[127] D. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.

[128] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[129] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[130] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proc. Conference on Empirical Methods in NLP*, pages 404–411, 2004.

[131] B. Hättasch, N. Geisler, C. M. Meyer, and C. Binnig. Summarization beyond news: The automatically acquired fandom corpora. In *Proceedings of The 12th Language Resources and Evaluation Conf.*, pages 6700–6708, 2020.

[132] C.-Y. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 45–51, 2002.

[133] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of ACL Conference*, pages 4171–4186, 2019.

[134] F. Alimoğlu and E. Alpaydin. Combining multiple representations for pen-based handwritten digit recognition. *Turkish Journal of Electrical Engineering & Computer Sciences*, 9(1):1–12, 2001.

[135] P. Sambaturu, A. Gupta, I. Davidson, S. S. Ravi, A. Vullikanti, and A. Warren. Efficient algorithms for generating provably near-optimal cluster descriptors for explainability. In *34th AAAI*, pages 1636–1643. AAAI Press, 2020.

[136] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of ACM KDD*, pages 226–231, 1996.

[137] C-D. Wang, J.-H. Lai, C. Y. Suen, and J-Y. Zhu. Multi-exemplar affinity propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(9):2223–2237, 2013.

[138] Michael Livanos and Ian Davidson. Identification and uses of deep learning backbones via pattern mining. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 697–705. SIAM, 2024.

[139] Gary B. Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild. workshop on faces in 'real-life' images. *FFINRIA*, 2008.

[140] Dan Stowell, Michael D Wood, Hanna Pamuła, Yannis Stylianou, and Hervé Glotin. The first bird audio detection challenge. *Methods in Ecology and Evolution*, 10(3):368–380, 2019.

[141] Oxford English Dictionary. Dog.

[142] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[143] Jianfei Zhu Gosta Grahne. Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions On Knowledge And Data Engineering*, 17, 2005.

[144] Walter A Kosters, Wim Pijls, and Viara Popova. Complexity analysis of depth first and fp-growth implementations of apriori. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 284–292. Springer, 2003.

[145] Thomas Grill and Jan Schlüter. Two convolutional neural networks for bird detection in audio signals. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1764–1768. IEEE, 2017.

[146] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey, 2020.

[147] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[148] Adam White and Artur d'Avila Garcez. Measurable counterfactual local explanations for any classifier, 2019.

[149] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): towards medical XAI. *CoRR*, abs/1907.07374, 2019.

[150] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[151] Jiaxwei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, Oct 2019.

[152] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks.

[153] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017.

[154] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.

[155] K. Crenshaw. Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics, 1989.

[156] I. Davidson, Z. Bai, and CM Tran. Making clusterings fairer by post-processing: algorithms, complexity results and experiments. *Data Min Knowl Disc*, 2022.

[157] C. Wadsworth, F. Vera, and C. Piech. Achieving fairness through adversarial learning: an application to recidivism prediction, 2018.

[158] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi. A survey on datasets for fairness-aware machine learning. *WIREs Data Mining and Knowledge Discovery*, 12(3), mar 2022.

[159] Dongxu Huang, Dejun Mu, Libin Yang, and Xiaoyan Cai. Codetect: Financial fraud detection with anomaly feature detection, 2018.

[160] Mohamad Zamini and Seyed Mohammad Hossein Hasheminejad. A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare, 2019.

[161] Rose Yu, Huida Qiu, Zhen Wen, ChingYung Lin, and Yan Liu. A survey on social media anomaly detection, 2016.

[162] David Savage, Xiuzhen Zhang, Xinghuo Yu, Pauline Chou, and Qingmai Wang. Anomaly detection in online social networks, 2014.

[163] Weijia Zhang and Xiaofeng He. An anomaly detection method for medicare fraud detection, 2017.

[164] Richard A Bauder and Taghi M Khoshgoftaar. Multivariate anomaly detection in medicare using model residuals and probabilistic programming, 2017.

[165] Clare Garvie, Alvaro Bedoya, and Jonathan Frankle. The perpetual line-up: Unregulated police face recognition in america, 2016.

[166] Denise Almeida, Konstantin Shmarko, and Elizabeth Lomas. The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of us, eu, and uk regulatory frameworks. *AI and Ethics*, 2(3):377–387, 2022.

[167] Hongjing Zhang and Ian Davidson. Towards fair deep anomaly detection, 2021.

[168] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets, 2014.

[169] Ge Zhang, Tianxiang Luo, Witold Pedrycz, Mohammed A El-Meligy, Mohamed Abdel Fattah Sharaf, and Zhiwu Li. Outlier processing in multimodal emotion recognition, 2020.

[170] Ranlei Cao, Xinyu Liu, Ju Zhou, Dong Chen, Dairong Peng, and Tong Chen. Outlier detection for spotting micro-expressions, 2021.

[171] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias, 2016.

[172] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: evaluating claims and practices, 2020.

[173] Ian Davidson and Selvan Suntiha Ravi. A framework for determining the fairness of outlier detection, 2020.

[174] Savitha Sam Abraham et al. Fairlof: fairness in outlier detection, 2021.

[175] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. Fairod: Fairness-aware outlier detection, 2021.

[176] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review, 2021.

[177] Geoffrey E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234, 1989.

[178] Nathalie Japkowicz, Colin Myers, and Mark A. Gluck. A novelty detection approach to classification, 1995.

[179] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification, 2018.

[180] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis, 2016.

[181] Hanyu Song, Peizhao Li, and Hongfu Liu. Deep clustering based fair outlier detection, 2021.

[182] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact, 2015.

[183] Sahil Verma and Julia Rubin. Fairness definitions explained, 2018.

[184] Harini Suresh and John Guttag. A framework for understanding sources of harm throughout the machine learning life cycle, October 2021.

[185] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification, 2009.

[186] B. Lingenfelter, S.R. Davis, and E.M. Hand. A quantitative analysis of labeling issues in the celeba dataset, 2022.

[187] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2015.

[188] The 88th United States Congress. Civil rights act of 1964. U.S. Statutes at Large, 1964. Public Law 88-352, 78 Stat. 241.

[189] The 90th United States Congress. Age discrimination in employment act of 1967. U.S. Statutes at Large, 1967. Public Law 90-202, 81 Stat. 602.

[190] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness, 2017.

[191] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification, 2018.

[192] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning, 2018.

[193] Berk Ustun, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees, 2019.

[194] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. Does mitigating ml's impact disparity require treatment disparity?, 2018.

[195] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2018.

[196] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9, 2010.

[197] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 2004:1–123, 2004.

[198] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerische Mathematik*, 93(2):223–256, 2002.

[199] Linda F. Wightman. LSAC National Longitudinal Bar Passage Study. LSAC Research Report Series., 00 1998. Report.

[200] PyTorch Contributors. torch.nn.linear, 2022. Accessed: 2024-05-18.

[201] Toon Calders and Indre Zliobaitė. Why unbiased computational processes can lead to discriminative decision procedures, 2013.

[202] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets, 2017.

[203] Razieh Nabi and Ilya Shpitser. Fair inference on outcomes, 2018.

[204] Usman Gohar and Lu Cheng. A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, August 2023.

[205] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments, 2017.

[206] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage, 1990.

[207] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

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

[209] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[210] G. E. Blelloch, H. V. Simhadri, and K. Tangwongsan. Parallel and I/O efficient set covering algorithms. In G. E. Blelloch and M. Herlihy, editors, *24th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '12, Pittsburgh, PA, USA, June 25-27, 2012*, pages 82–90. ACM, 2012.

[211] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983, 2019.

[212] G. B Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. Tech. Report 07-49, UMass Amherst, 2008.

[213] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.