Understanding Biomedical Machine Learning Models

Joseph D. Janizek

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Su-In Lee, Chair

Linda Shapiro

Marshall Horwitz

Program Authorized to Offer Degree:

Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

Understanding Biomedical Machine Learning Models

Joseph D. Janizek

Chair of the Supervisory Committee:
Professor Su-In Lee
Paul G. Allen School of Computer Science and Engineering

As complex, black box models have increasingly come to predominate the algorithms used in state-of-the-art machine learning pipelines, the need to explain and understand the predictions made by these algorithms has grown correspondingly. *Feature attribution methods* are one popular approach to explain these black box models, but are limited in their expressive capacity. We therefore propose three approaches to go beyond the shortcomings of existing feature attribution methods. The first, EXPRESS, demonstrates how the stability and quality of feature attributions for models of gene expression data increase when these models are ensembled. The second, Integrated Hessians, efficiently explains the interactions between pairs of features for neural network models, which we show has general applications even beyond biological and medical models. In a third approach, we apply generative adversarial networks and saliency maps to identify the underlying reasons for poor generalizability of radiographic COVID-19 detection models. Furthermore, while the utility of feature attribution methods for helping humans understand what models have learned is well-known, their utility for helping humans express their own desiderata in machine-interpretable language is under-appreciated. We develop a feature attribution method that is designed for use during model training, and demonstrate how it can be used to incorporate gene interaction networks as a constraint on predictive models with gene expression features. Finally, we show how to enforce more abstract model constraints using adversarial training in the context of radiographic pneumonia classification.

TABLE OF CONTENTS

DEDICATION

To everyone without whom this thesis would not have been possible: God, my family (including my fiancée Samantha, my parents Patricia and David, my brother John, and all of the other Clarks/Janizeks/Gilberts), and my friends (who, despite being Knuckleheads, have never allowed me to forget that I am money)

Chapter 1

INTRODUCTION

In recent years, machine learning models have demonstrated significant performance improvements across many domains. In the area of Natural Language Processing, large language models have achieved state-of-the-art performance at tasks ranging from automatic speech recognition, to machine translation, to document classification [29]. Likewise, in Computer Vision, models have proved useful in applications as diverse as object detection for self-driving vehicles [229], molecular marker status determination in digital pathology pipelines [197], and lung cancer screening with computed tomography images [6]. One commonality between all of these successful applications is that they tend to employ complex, *black box* algorithms — models for which the reasoning underlying predictions is opaque.

The black box nature of these algorithms poses a variety of problems. Despite the apparently high predictive performance attained by models, deep learning algorithms are often brittle, meaning that their performance substantially degrades in response to real world distribution shifts [133]. This brittleness is likely due to "shortcut learning," where models learn training set-specific factors rather than robust and general patterns [79]. Auditing models to understand whether or not they rely on plausible mechanisms may be difficult or impossible in the context of black box models. In addition to the possible reliance on shortcuts, explanations are sometimes legally required for applications like credit scores and loan underwriting.[1] Finally, in some domains, like genomics, elucidating the underlying mechanism of some process may be more interesting than accurately predicting that process [9].

The work presented in this thesis is motivated by the specific challenges that arise when trying to understand biological and medical machine learning models. The structure is broken down into five main chapters, each corresponding to a first author paper written during my graduate education [113, 66, 114, 52, 112]. At a high level, the first three chapters after the introduction (Chapters 2-4) involve *trying to understand what models have learned* through the development and application of explainable AI methods. The last two chapters (Chapters 5-6) involve developing methods to *communicate domain knowledge back to complex models*, thereby improving their performance.

---

[1] e.g., Equal Credit Opportunity Act (Regulation B of the Code of Federal Regulations), Title 12, Chapter X, Part 1002, §1002.9.

One major class of methods for understanding what models have learned is *feature attribution* [179]. These methods explain the predictions of a function by producing scalar attributions for each of the function's input features. These attributions may be global, indicating the importance of the feature over all samples in a dataset, or local, indicating the importance of the feature for a particular sample [46, 177]. Methods for calculating feature attribution include both Shapley value-based algorithms [45], and gradient-based approaches [274, 262].

Trying to use feature attributions to analyze the transcriptomic factors underlying anti-cancer drug synergy presents a unique challenge, however. Using a variety of synthetic datasets, we benchmark both classical and novel approaches for discovering biologically-relevant features from gene expression data and demonstrate how non-linearity and correlation in these data impede feature attribution methods (**Chapter 2**). We then demonstrate that under conditions representative of typical biological applications, all feature attribution approaches tend to perform poorly, and show how explaining ensembles of models improves the quality of feature attributions.

Additionally, despite the utility of feature attributions, there is significant interest in more detailed explanations. For example, a recent qualitative study of machine learning engineers noted the need for methods not only to understand the isolated importance of individual features, but also to quantify the interactions *between* pairs of features [19]. Hence, we propose Integrated Hessians (**Chapter 3**)[2], a method for detecting and describing interactions between features. Additionally, while feature attribution methods work well on tabular datasets where features have intrinsic meaning, for many image datasets, models may rely on complex patterns that are difficult to represent in terms of static pixels. We therefore propose using generative models in addition to saliency maps to understand important differences between classes of images, and use this to help audit radiographic COVID-19 detection models (**Chapter 4**)[3].

While explainable machine learning approaches are well-known to help users understand what black box models have learned, it is significantly less appreciated that these methods may also allow users to express their domain knowledge in a language understandable by the model. Hence, we propose attribution priors, a framework for regularizing the explanations of differentiable models, and expected gradients, an attribution method designed for efficient incorporation into the model training process (**Chapter 5**)[4]. While part of the benefit of deep learning models is their ability to automatically learn useful feature representations without the need for hand-crafted features, domain experts often have knowledge that can be helpful to incorporate into the modeling process. For example, in biology there are often

[2]This work was co-first authored with Pascal Sturmfels.

[3]This work was co-first authored with Alex DeGrave.

[4]This work was co-first authored with Gabe Erion and Pascal Sturmfels.

"meta-features," or multi-modal heterogeneous data that it is desirable to incorporate into models [157]. While this has traditionally been straight-forward to integrate as a constraint on the parameters of linear methods, extending this approach to deep networks is non-trivial.

In the same way that feature attribution methods struggle to describe the complex patterns on which image models rely in terms of static pixels, incorporating domain knowledge into image models can often be difficult to specify as a prior on feature attributions. Telling a model not to use complex combinations of features, textures, etc., is nearly impossible to state as a function of individual pixels. This is particularly relevant in the context of medical imaging models, where brittleness is often a problem. For example, [305] found that a deep learning pneumonia classifier trained on data from two hospital systems exploited differences in the base rate of pneumonia between the two hospitals by learning to identify each radiograph's hospital of origin rather than anatomically-relevant features of pneumonia. While this model apparently had high predictive performance, when the model was tested on radiographs from a third hospital not present in the training data its performance significantly decreased. Furthermore, even within a single hospital system, confounded predictions may be a problem for deep learning. For example, [11] demonstrated that a deep learning hip fracture classifier was leveraging patient-level variables (such as age and gender) and process-level variables (such as scanner model and hospital department) in its predictions. After controlling for these variables during model evaluation by rebalancing the test set, they found that the classifier performed no better than random. While the works above have described the brittleness of deep learning medical imaging classifiers, more work is needed to create robust models. We therefore propose an adversarial approach to improve pneumonia models (**Chapter 6**), and show how it improves generalization.

Chapter 2

UNCOVERING EXPRESSION SIGNATURES OF SYNERGISTIC DRUG RESPONSE USING AN ENSEMBLE OF EXPLAINABLE AI MODELS

2.1 Introduction

Acute myeloid leukemia (AML) is the most commonly diagnosed form of leukemia in adults, and carries a poor prognosis [124]. While survival has improved over the past several decades for younger patients, older patients have not seen a similar improvement. This gap in survival has motivated the development of molecularly targeted combination therapies for patients who do not qualify for intensive induction chemotherapy [147]. Discovering optimal combinations of anti-cancer drugs is a difficult problem, however, as the space of all possible combinations of drugs and patients is large. While potentially synergistic drug combinations have traditionally been tested on the basis of either biological or clinical expert knowledge [51], more systematic approaches are necessary to effectively explore this space. Even systematic experimental approaches such as high-throughput screening are potentially insufficient, as there are hundreds of thousands of possible combinations of all anti-cancer drugs currently in development, each of which may have a different response in different patients [51, 208]. Therefore, predictive approaches are necessary to make the immense space of possible anti-cancer drug combinations manageable.

State-of-the-art predictive approaches fall short along another axis, however, by failing to provide biological insight into the molecular mechanisms underlying drug response, which is essential to facilitate the discovery of new and effective anti-cancer therapies [116, 198, 286]. While a wide variety of computational methods have historically been employed for drug combination prediction [245, 102, 35, 69, 299], recent work has demonstrated increased predictive performance using complex, non-linear machine learning (ML) models. For example, all of the winning teams in the AstraZeneca-Sanger Drug Combination Prediction DREAM Challenge utilized complex models in some part of their approach, including ensembles of random forest classifiers and gradient boosted machines [190]. Additionally, Preuer et al. have shown that deep neural networks outperform less sophisticated models such as linear models, achieving state-of-the-art performance at predicting the synergy of anti-cancer drug combinations in 39 cell lines [220]. A major weakness of these complex ML models is their "black box" nature; despite their high predictive accuracy, these models' inner workings

are opaque, making it challenging to gain mechanistic insights into the molecular basis of drug synergies. In cases where model interpretability is important, researchers resort to simpler, less accurate models like linear regression. For example, to identify genomic and transcriptomic markers associated with drug sensitivity, both the Cancer Genome Project [78] and the Cancer Cell Line Encyclopedia [14] used penalized elastic net regression.

Here, we present the EXPRESS (**ex**plainable **pr**edictions for gene expr**ess**ion data) framework to understand the relationship between accuracy and interpretability in biological models, and build models that are both accurate *and* biologically interpretable. A recent approach to understand the patterns learned by biological models involves "explaining" complex predictive models using *feature attribution methods*, like Shapley values [179, 177, 256, 274], to provide an importance score for each input feature (here, a gene). The Shapley value is a concept from game theory designed to fairly allocate credit to players in coalitional games [252]. By considering input features as players and the model's output as the reward to be allocated [179], the most important features can be identified for complex models that would otherwise be uninterpretable. Unfortunately, the application of off-the-shelf feature attribution methods is unlikely to be successful in the context of large cancer 'omics data. These methods are known to struggle in the setting of high-dimensional and highly correlated features, such as those present in transcriptome-wide gene expression measurements [1]. Furthermore, while complex ML models have been shown to achieve increased predictive performance when compared to simpler models, recent work has raised the concern that models with higher predictive performance do not necessarily have higher-quality attributions on the same tasks [137, 247]. Our results investigate the relationship between predictive performance and feature attribution quality, and demonstrate how a simple approach based on model ensembles can improve the feature attribution quality of complex machine learning models in the life sciences.

First, using 240 synthetic datasets, we benchmark both classical and novel approaches and demonstrate how non-linearity and correlation in the data can impede the discovery of biologically relevant features. We then demonstrate that under conditions representative of typical biological applications, all existing approaches tend to perform poorly, and show how explaining ensembles of models improves the quality of feature attributions (Fig. 2.1a). Finally, we describe EXPRESS, which uses Shapley values to explain an *ensemble* of complex models trained to predict drug combination synergy on a dataset of 133 combinations of 46 anti-cancer drugs tested in *ex vivo* tumor samples from 285 patients with AML (Fig. 2.1b, Fig. 2.7). In addition to building highly accurate predictive models, our ensemble interpretability approach identifies relevant biological signals underlying drug synergy patterns, most notably a gene expression signature related to hematopoietic differentiation. While individualized treatment for AML on the basis of cancer genomic signatures is already becoming an important aspect of clinical practice [198], our approach identifies a novel *expression*-based signature

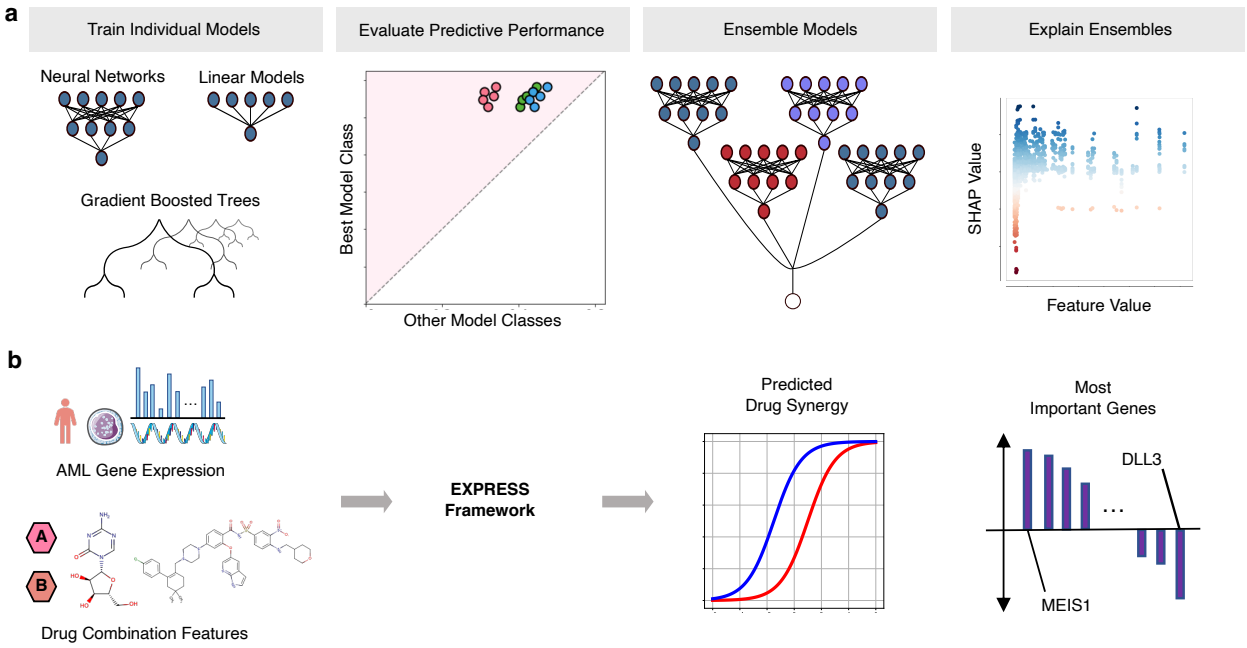

Figure 2.1: **Overview of the study design. a**, Our framework, EXPRESS (explainable predictions for gene expression data), for learning reliable explanations of cancer therapeutic machine learning models trained on high-dimensional gene expression data. After training a variety of individual models across multiple model classes, predictive performance is evaluated to select a best-performing model class. Multiple models from that class are then ensembled in order to produce more reliable and biologically-meaningful explanations. **b**, We apply our pipeline to a dataset of *ex vivo* anti-cancer drug synergy measurements for patients with acute myeloid leukemia, attaining not only superior prediction performance, but also identifying biological processes that are important for the determination of drug synergy.

that is predictive of synergy across a broad class of drugs and their combinations in AML.

2.2 Results

2.2.1 Current state-of-the-art explainable AI falls short on correlated features

Explainable AI (XAI) is a recent development in the ML community that attempts to provide a human-interpretable basis for the predictions of complex, "black box" models like neural networks. In particular, feature attribution methods are a class of methods that identify the relative importance of each input feature (e.g. the expression level of a gene) for a particular model [179, 45]. One popular feature attribution method involves applying Shapley values

to interpret these complex models by measuring how much the model's output changes on average when a feature is added to all other possible coalitions of features (Methods 2.4.1).

While applying explainable AI techniques to complex models has become a popular practice in the life sciences [127, 126, 248, 107, 189, 12, 234, 233, 281, 24], applying these methods in the context of gene expression data is particularly difficult. Each patient will have a transcriptomic profile with tens of thousands of features with a high degree of feature interdependence (e.g., see the feature covariance matrix for AML transcriptomic data, Fig. 2.2, top right). This makes the task of accurate feature attribution harder for Shapley value algorithms, which ideally would operate on statistically independent features [179]. In the presence of correlated features, many models with diverse mechanisms could potentially fit the data equivalently well [27, 60]. Thus, even if we could explain a single model perfectly, that model might not correspond well to the true biological relationships between features and outcome.

Since these conditions are ubiquitous in biological datasets, it is essential to understand how the efficacy of both Shapley value-based attributions and more conventional methods will be impacted in the setting of high-dimensional, highly correlated features. Measuring this efficacy is difficult, however, as existing benchmarks of feature attribution methods are designed to either measure the influence of features on the *particular model* being explained [177], or to measure the *predictive performance* of selected sets of features [101]. We therefore design a simple benchmark for this application (Fig. 2.2, Methods 2.4.2). To evaluate the effects of data correlation and non-linearity on feature attribution, we use 240 unique datasets. As input data, we consider synthetic datasets with independent features and synthetic datasets with multivariate normal covariance structure, as well as datasets with real gene expression measurements sampled from AML patients [286]. Since the goal of our benchmark is to define how well different methods recover *true features*, we create synthetic labels, allowing the ground truth to be recovered and measured. These labels are created by randomly sampling input features and relating them to the outcome using functions ranging from simple linear, univariate relationships to complex, non-linear step functions with interactions between features (Methods 2.4.2). For our metric of feature discovery performance, we measure how many *true features* are found cumulatively at each point in the lists of features ranked by each feature attribution method (Methods 2.4.2, Fig. 2.8). Using this benchmark, we then evaluate five different methods for ranking biologically important features, including two complex machine learning methods (gradient boosted machines, neural networks) explained using Shapley values, as well as three more traditional linear methods: ranking features by their Pearson correlation with the outcome [264], ranking features by their elastic net coefficients [315], and recursive feature elimination using support vector machines [94].

When the outcome has a simple linear relationship with the input features, all approaches recover the true features well (see the perfect performance across all methods in the top

left experiment in Fig. 2.2). When there is non-linearity in the data, however (see bottom two rows of Fig. 2.2), the complex machine learning models interpreted with Shapley values significantly outperform the linear approaches. For example, neural networks explained with Shapley values attain a higher AUFDC than elastic net coefficients when the true outcome is multiplicative and the features are independent (Mann-Whitney U-test, $p = 3.3 \times 10^{-6}$, $U = 4.65$) or in correlated groups ($p = 6.3 \times 10^{-8}$, $U = 5.41$). Likewise, XGBoost models explained with Shapley values attain a higher AUFDC than elastic net coefficients when the true outcome is a pairwise-AND function and the features are independent ($p = 6.5 \times 10^{-7}$, $U = 4.98$) or in correlated groups ($p = 3.7 \times 10^{-7}$, $U = 5.09$). Importantly, however, as the correlation between input features increases to the level seen in real AML transcriptomic data (Fig. 2.2, third column), all methods tend to perform poorly and there is a high degree of variance in the performance of each model class.

2.2.2 Ensembling overcomes variability in individual models

Given the observed variability of different models in terms of benchmark performance, a natural question that arises is how to select the predictive model that will attain the best performance at feature discovery. An intuitive solution is to simply pick the model with the best predictive performance. When we examine the relationship between predictive performance and feature discovery, however, we see that this is not necessarily a reliable strategy. For each of three popular model classes (linear models, feed-forward neural networks, and gradient boosted machines), we train twenty independent models on bootstrap resampled versions of the same dataset and measure test set prediction error and feature discovery performance. While there was significant overall correlation between test error and feature discovery (Step Function Dataset: Pearson's $r = -0.77$, $p = 1.1 \times 10^{-12}$, $n = 60$; Multiplicative Dataset: Pearson's $r = -0.82$, $p = 1.2 \times 10^{-15}$, $n = 60$), within each model class test error was *not* significantly correlated with feature discovery performance (Elastic Net + Step Function Dataset: Pearson's $r = 0.19$, $p = 0.43$, $n = 20$; Neural Network + Step Function Dataset: Pearson's $r = 0.02$, $p = 0.94$, $n = 20$; XGBoost + Step Function Dataset: Pearson's $r = -0.18$, $p = 0.45$, $n = 20$; Elastic Net + Multiplicative Dataset: Pearson's $r = -0.11$, $p = 0.65$, $n = 20$; Neural Network + Multiplicative Dataset: Pearson's $r = -0.22$, $p = 0.35$, $n = 20$; XGBoost + Multiplicative Dataset: Pearson's $r = 0.13$, $p = 0.60$, $n = 20$; see Fig. 2.3a,b). Therefore, while predictive performance may help to select a *model class*, it will not necessarily help to select which model within that class has the most biologically-relevant explanations.

Furthermore, when we examine the feature attributions across individual models within a single model class, we observe that they vary substantially from model to model (Fig. 2.10). This indicates a lack of stability in the attributions: minor perturbations to the training set

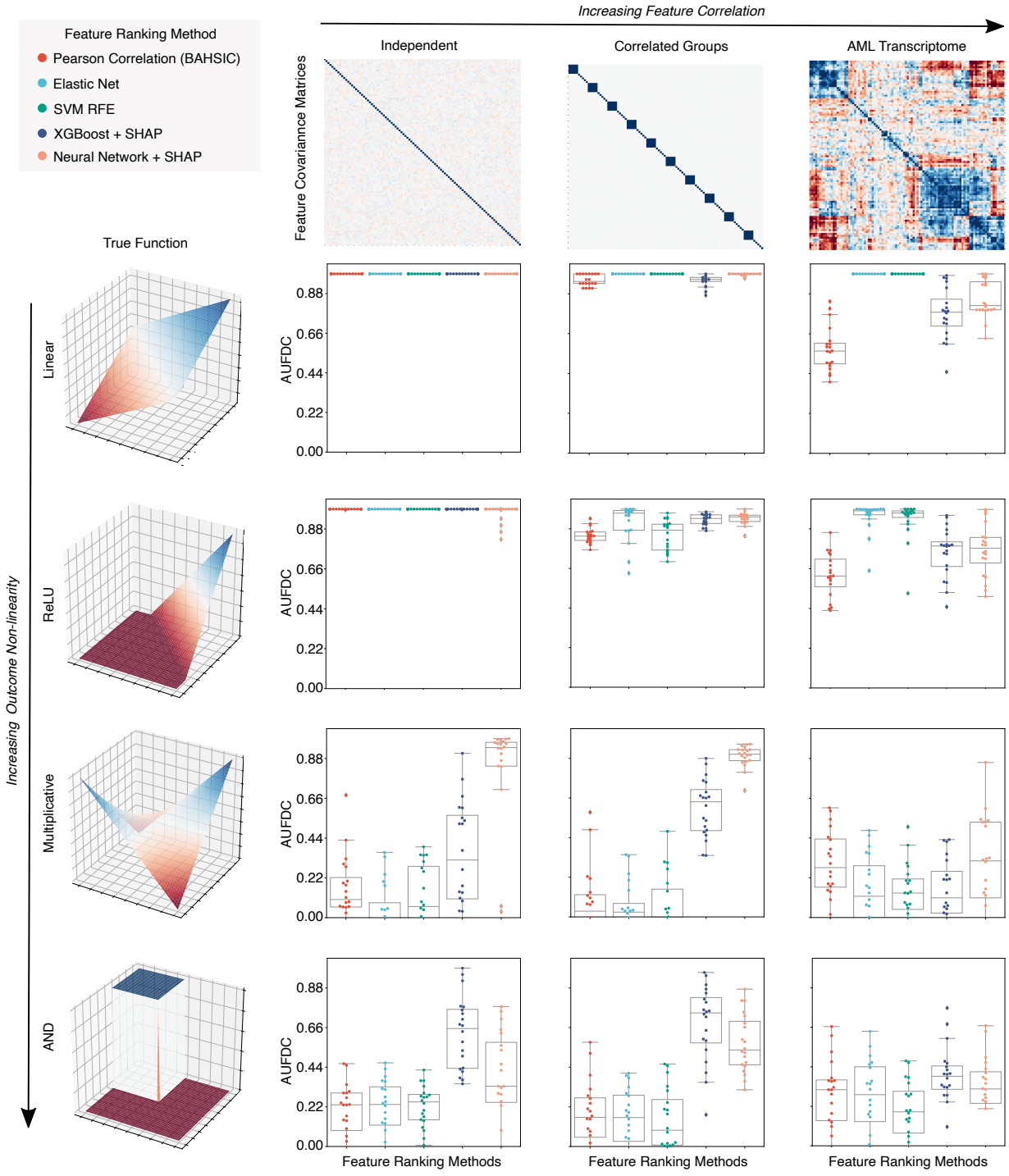

Figure 2.2: **Benchmark metric reveals the impact of non-linearity and correlation on feature discovery.** Each point in the box plots represents the benchmark score achieved by one of five feature ranking methods applied to one of 240 datasets generated from twelve synthetic or semi-synthetic dataset types (each subplot represents one dataset type). The rows (left) are sorted from top to bottom by increasing non-linearity of the true feature-outcome relationship (continued on next page)

Figure 2.2: (Previous page.) (e.g. all datasets in the first row have a linear relationship between input features and outcome) while the columns are sorted from left to right by the increasing extent of the correlation between features in the dataset (e.g. all datasets in the last column have real AML bulk RNA-seq features). The metric plotted in each boxplot is the area under the feature discovery curve (AUFDC) (see Methods 2.4.2), where a higher score indicates better performance (0 represents random performance, while 1 represents perfect performance). The boxes mark the quartiles (25th, 50th, and 75th percentiles) of the distribution, while the whiskers extend to show the minimum and maximum of the distribution (excluding outliers). For each dataset type (a pair of feature-outcome relationship and inter-feature correlation), 20 independent datasets are generated by randomly regenerating features. While all approaches achieve perfect performance on simple linear data with independent features (top left plot), all models have worse performance as features become more correlated and outcomes become more non-linear (bottom right)

(such as bootstrap resampling) can lead to substantial variability in the features identified as most important by the model [27], and prior work in machine learning applied to human genomics and epigenomics has suggested the necessity of considering multiple models when analyzing explanations [10, 188]. Likewise, recent work on feature selection for black box predictive models in healthcare has pointed out the need to select robust features [68].

While ensembling machine learning models is classically known to increase the accuracy of models by increasing stability of predictors, it remains to be demonstrated whether ensembling can improve biological hypothesis generation. We therefore created ensembles of models for all of the datasets in the original benchmark task, and found that ensembling not only decreases the variance in feature discovery performance, but also significantly increases the average feature discovery performance of the ensemble models (Fig. 2.3c). Not only does this improvement occur consistently across dataset types and model classes (see Fig. 2.11-2.12), but this effect is independent of an increase in predictive performance (see Fig. 2.13). Furthermore, this effect is greater than that seen by adding explicit regularization to models (see Fig. 2.14).

To understand how the ensembled models differed from the individual models, we analyzed the difference between the attributions attained by a variety of ensembled models and the individual models. We see that the variability in attributions across bootstrap resampled versions of the dataset decreases (with an average pairwise cosine similarity between attributions across models increasing from 0.77 to 0.98 after ensembling, $p = 4.87 \times 10^{-63}, U = 16.76$, Fig. 2.10). Furthermore, when compared to the single models, the ensemble models tend

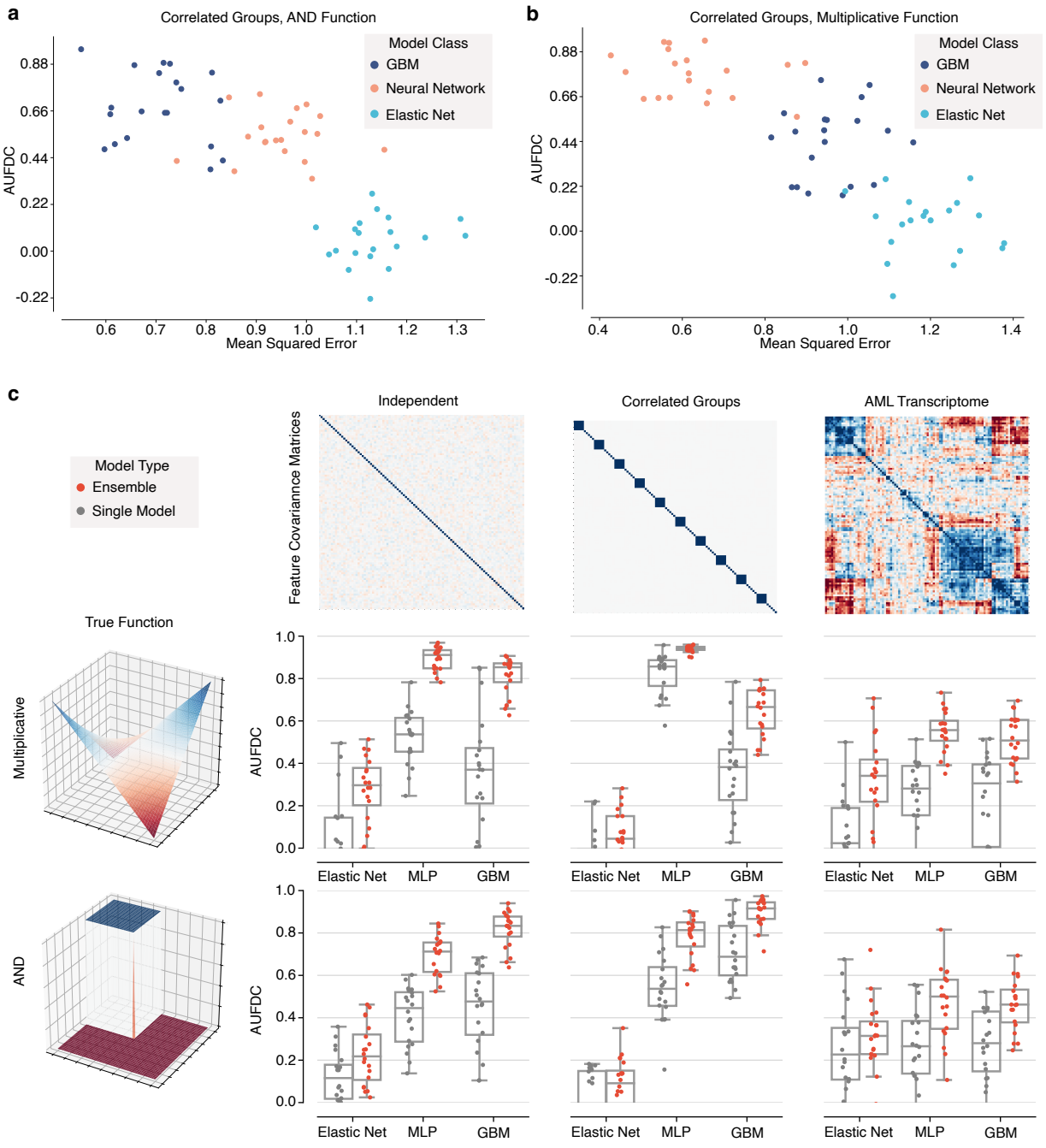

Figure 2.3: **Explaining ensembles helps overcome instability in feature discovery performance for single models. a-b**, Relationship between test error and feature discovery performance on bootstrap resampled versions of two synthetic datasets. Both datasets had clusters of highly-correlated features; one had a step function outcome (**a**) while the second had a multiplicative outcome (**b**). While there is a high overall correlation between test error and feature discovery performance for both datasets (Continued on next page.)

Figure 2.3: (Previous page.) (Step Function Dataset: Pearson's $r = -0.77$, $p = 1.1 \times 10^{-12}$, $n = 60$; Multiplicative Dataset: Pearson's $r = -0.82$, $p = 1.2 \times 10^{-15}$, $n = 60$), there is no significant correlation after conditioning on model class (Elastic Net + Step Function Dataset: Pearson's $r = 0.19$, $p = 0.43$, $n = 20$; Neural Network + Step Function Dataset: Pearson's $r = 0.02$, $p = 0.94$, $n = 20$; XGBoost + Step Function Dataset: Pearson's $r = -0.18$, $p = 0.45$, $n = 20$; Elastic Net + Multiplicative Dataset: Pearson's $r = -0.11$, $p = 0.65$, $n = 20$; Neural Network + Multiplicative Dataset: Pearson's $r = -0.22$, $p = 0.35$, $n = 20$; XGBoost + Multiplicative Dataset: Pearson's $r = 0.13$, $p = 0.60$, $n = 20$;). **c**, Comparison of feature discovery performance between individual models and ensemble models synthetic and semi-synthetic datasets from our benchmark. The boxes mark the quartiles (25th, 50th, and 75th percentiles) of the distribution, while the whiskers extend to show the minimum and maximum of the distribution (excluding outliers). Results for the rest of the datasets and for additional feature attribution methods can be found in Figs. 2.11-2.12.

to place more weight on a small set of important features and attribute less importance to spurious correlates: spurious correlations cancel out over repeated model trainings, while true signal remains consistent (Fig. 2.10). In addition to carrying out this experiment for the other two "True Functions" evaluated in the original benchmark in Fig. 2.2, we also verified that the improvement seen with ensembling holds when other feature attribution methods like DeepLift [256] and Integrated Gradients [274] are used (Figs. 2.11-2.12).

These results suggest a natural approach for applying explainable AI techniques to complex biological datasets (see Fig. 2.1a). A variety of model classes should be compared in terms of predictive performance, and following the selection of the best performing model class, the set of well-performing models from that class should be ensembled for explanation.

2.2.3 Complex gradient boosted machines accurately predict drug synergy in AML samples

After determining the importance of model class selection and model ensembling from our benchmark, we applied our framework to publicly available data provided by the Beat AML collaboration [286]. These data consist of the gene expression profiles of primary tumor cells from 285 patients with acute myeloid leukemia, as well as drug synergy measured for these cells in an *ex vivo* sensitivity assay for 131 pairs of 46 distinct drugs, spanning a variety of cancer subtypes and anti-cancer drug classes (Fig. 2.7). The input features of each sample thus comprise 'gene expression features' which describe the corresponding patient's tumor's molecular profile, and 'drug features' which describe the two drugs in that combination in terms of the gene targets of each of the two drugs (Fig. 2.1b and Methods).

EXPRESS begins by comparing multiple model classes: elastic net [315], deep neural networks [220], random forests [26], and extreme gradient boosting (XGBoost) [39], in terms of the test error calculated using 5-fold cross validation tests. To rigorously evaluate the predictive performance of the models, we performed comparisons using four different schemes for stratifying samples into train and test sets. Each different stratification assesses the generalization performance for a different possible application scenario (see Fig. 4 and Methods) [220, 130]. Across these four settings, XGBoost shows better performance in 53 comparisons out of 60 (=4×3×5) comparisons from four settings, with three alternative methods, and for five test folds. Elastic net, random forests, and deep neural networks show better performance in 4, 27, and 30 comparisons, respectively. Our framework therefore selects XGBoost as the optimal model class for further downstream interpretive analysis. This finding aligns well with contemporary work on machine learning for tabular datasets (like gene expression data), which has empirically demonstrated that tree models like XGBoost tend to outperform deep learning models [257].

2.2.4 Ensembled attributions reveal important genes for anti-AML drug synergy

After identifying gradient boosted machines as the best-performing model class for our dataset, we ensembled individual models until the ensemble model attributions were stable, leading to a final ensemble of 100 XGBoost models (see Fig. 2.15, Methods). We then analyzed the resultant ensemble model attributions to look for genes with *global* importance for drug combination synergy, i.e., genes whose expression is related to synergy across many different drug pairs in our dataset [177]. Genes that impact global synergy could belong to pathways with outsize importance to cancer biology which are targeted by many drugs in the dataset, such as MAPK signaling or PI3K-Akt signaling, or could be related to larger-scale transcriptional changes impacting many pathways simultaneously, such as the degree of differentiation of leukemic cells [92].

We first visualize genes with monotonic relationships with synergy across all samples in the dataset (measured by the strength of the Spearman correlation between expression and attribution values) by plotting these robust attributions in a dependence plot. For example, a strong positive correlation between the expression level of MEIS1 (the 2nd strongest relationship out of 15,377 genes tested), and its attribution value indicates that patients with higher levels of MEIS1 expression are predicted to respond more synergistically to the drug pairs tested in this dataset (Fig. 2.5a). MEIS1 has been shown to be upregulated in mixed-lineage leukemia (MLL)-rearranged AML [144], while also driving leukemogenesis independently of MLL-rearrangement [167]. Recently, high MEIS1 expression has been observed within Venetoclax-resistant AML subclones with "monocytic" characteristics [214]. Because AML in different patients may manifest in different developmental stages [214],

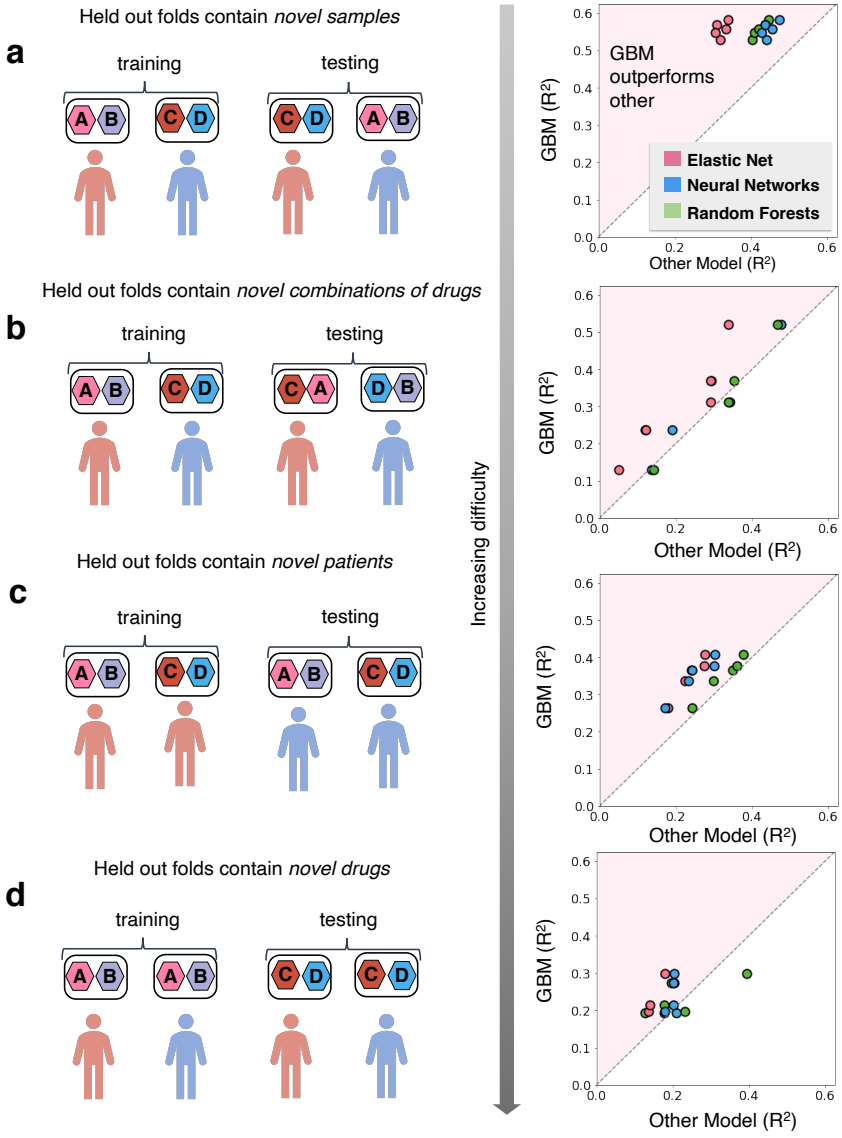

Figure 2.4: **Comparison of predictive performance between model classes across four stratification settings.** Each point in the plots on the right represents an evaluation of model performance after a different split of the data. In order to consider a variety of potentially useful application settings, samples were stratified in four ways. Each sample comprises primary tumor cells from a patient with AML and a pair of anti-cancer drugs. In **a**, samples are randomly split into 5 different train test folds. (Continued on next page.)

Figure 2.4: (Previous page.) In **b**, samples are split on the basis of the drug combinations, so that held out test folds contain novel drug combinations not present in the training data. In **c**, samples are split on the basis of patients, so that held out test folds contain patients not present in the training data. In **d**, samples are split on the basis of individual drugs, so that held out folds contain drugs not present in the training data.

the importance of MEIS1 suggests that our model may be learning a differentiation-related expression signature underlying the synergistic ability of certain drugs to overcome resistance to others.

EXPRESS can identify other genes showing such trends and visualize many of these feature attribution relationships at once by assembling the marginal distributions of the expression-attribution dependence plots into a summary plot. Figure 2.5c-d shows two summary plots – one for the genes where higher expression correlates with higher predicted synergy, and another for the genes with negatively correlated relationships. One of the top negatively correlated genes was DLL3 (Fig. 2.5b), a member of the Notch signaling pathway, which has been shown to have prognostic significance in patients with AML: patients with higher DLL3 expression have been shown to have lower overall survival [278]. We find that many of the top genes underlying synergy in both directions have been related to different stages of hematopoietic development. For example, CITED2 (the top positively correlated gene) is known to be essential for the maintenance of adult hematopoietic stem cells [140]. Additionally, CITED2-mediated hematopoietic stem cell maintenance has also been shown to be critical for the maintenance of AML [139]. Other genes in this list, such as OSMR, have further been shown to be essential for the maintenance of normal hematopoiesis [280]. Still other top genes, like SLC7A11 and SLC17A7, have been linked to prognosis of AML [311, 309, 162].

In addition to considering genes whose expression consistently impacts synergy either positively or negatively across all drug combinations, we additionally ranked genes by the magnitude of their global attribution values. This analysis allows genes that are important for multiple combinations to be ranked highly, even if higher expression of these genes are linked with higher synergy for some combinations and lower synergy for other combinations. When EXPRESS ranks all genes by the magnitude of their global attribution values (see Methods), we again find genes that are related to hematopoietic development and AML prognosis. In particular, IL-4 (top ranked gene by non-directional magnitude) is an important cytokine regulating the tumor microenvironment that has been shown to be specifically downregulated in AML compared to normal myeloid cells [138]. STAT6 (ranked 31st) is

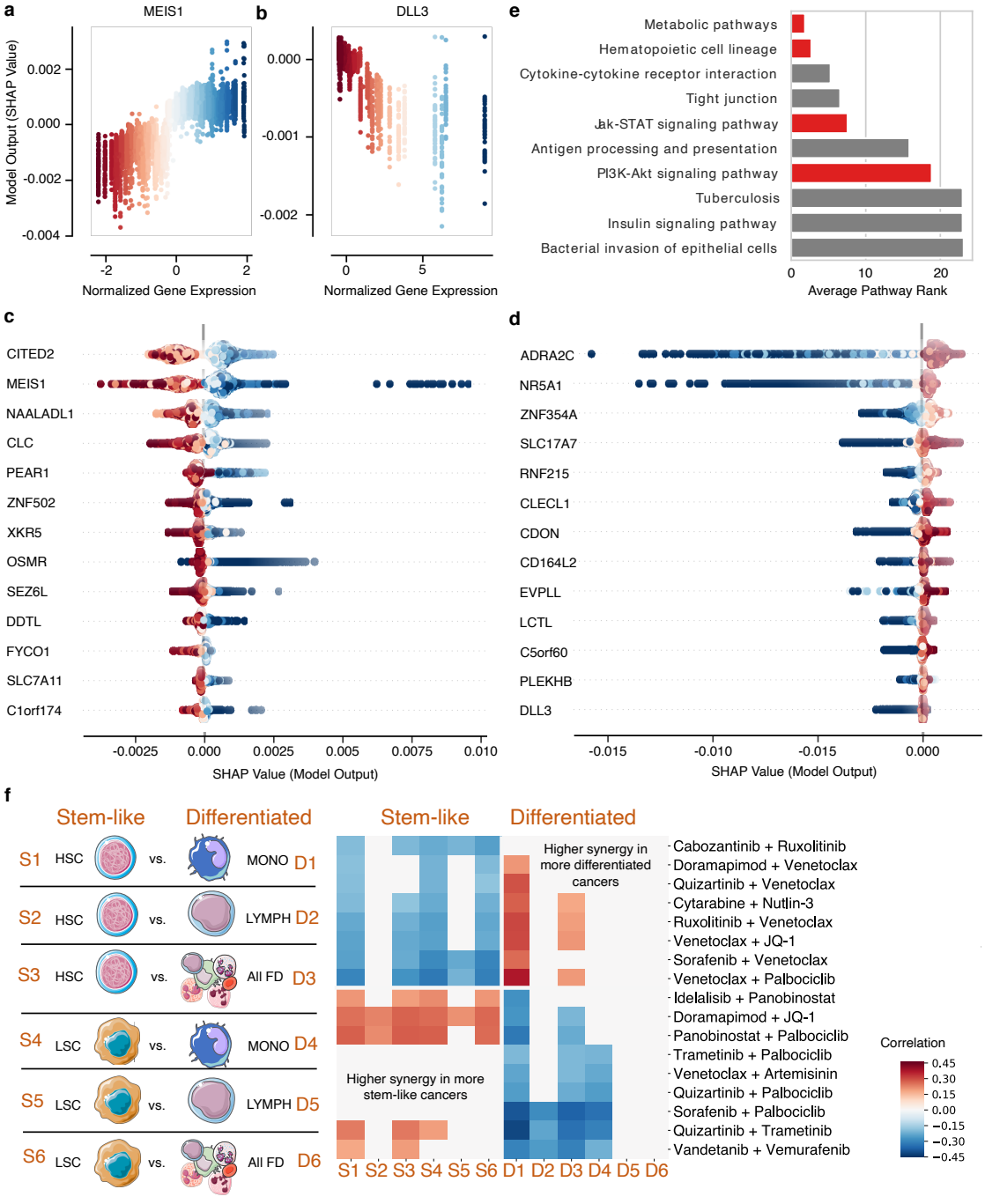

Figure 2.5: **Transcriptomic factors affecting anti-AML drug combination synergy. a-b**, SHAP dependency plots for MEIS1 and DLL3. Each point represents a single sample (one patient with a pair of anti-cancer drugs), the x-axis and color encoding represent the normalized gene expression values, while the y-axis represents the feature attribution value (continued on next page).

Figure 2.5: (Previous page.) (change in predicted drug synergy attributable to that feature). **c-d**, SHAP summary plots for the transcripts with the strongest positive (**c**) and negative (**d**) relationships with anti-AML drug synergy.Each point still represents a single sample and the color encoding still represents normalized gene expression values, while the x-axis now represents the feature attribution value (plotted on the y-axis in the corresponding analysis in **a** and **b**). **e**, Biological pathways most highly enriched in the list of most important gene expression features, sorted by their average ranking across several top gene thresholds. Red bars indicate pathways discussed further in the text. **f**, For twelve separate differential gene expression profiles created by pairing gene expression measurements from a more stem-like hematopoietic lineage cell population (HSCs or LSCs) with a more differentiated hematopoietic lineage cell population (monocytes, lymphocytes, or all fully differentiated cells), we measured the correlation between the average expression of that profile and the synergy for each drug combination. After FDR correction, we plotted all combinations with significant correlations across at least two profiles. We find that some combinations of drugs tend to have higher synergy in more differentiated cancers, while some combinations of drugs tend to have higher synergy in more stem-like cancers.

a transcriptional regulator known to be a key mediator of cytokine signaling [86]. It has previously been experimentally demonstrated using CRISPR-Cas9 genomic engineering that STAT6 specifically mediates IL-4-induced apoptosis in AML [215]. Furthermore, expression of STAT6 has been shown to be high in hematopoietic stem cells, but not in more differentiated progenitors [32]. Other top genes in this list, such as SLC51A and RNF213 (ranked 2nd and 6th overall, respectively), have been previously linked to AML and familial myelodysplasia via GWAS studies [161, 42].

2.2.5 Pathway explanations identify global importance of a differentiation signature

While attributions and trends for individual genes are informative, to gain systems-level insights into the processes important to drug synergy prediction, we can also use pathway databases to systematically check if genes from certain pathways are over-represented in EXPRESS's top-ranked genes. When we test the top-ranked genes for pathway enrichment, we find that the top pathway (Fig. 2.5e) is related to cellular metabolism. Expression programs regulating cancer metabolism have previously been linked to resistance to a variety of the drugs tested in this dataset. For example, AML cells that are resistant to the tyrosine kinase inhibitor Cabozantinib have been shown to have higher glucose uptake, GAPDH activity, and lactate production than Cabozantinib-sensitive cells [170].

Furthermore, consistent with our hypothesis that the importance of MEIS1 for synergy may be linked to a differentiation signature, the second most highly enriched pathway contains genes that control differentiation along the hematopoietic cell lineage ($p = 8.2 \times 10^{-3}$ from hypergeometric test, FDR-correction using Benjamini-Hochberg procedure.). Previous studies have shown that leukemic stem cell signatures associate with worse clinical outcomes [75, 81], and cells at different differentiation stages have been shown to respond differently to particular combination therapies [218, 149]. The differentiation signature and metabolic signature may in fact be related, as prior work has shown that less differentiated leukemic cells have unique metabolic dependencies [118], and have even proposed metabolic changes as a mechanism mediating anti-cancer drug combination resistance specifically in stem-like leukemic cells [266].

To further explore the importance of differentiation signatures as a global pattern underlying drug combination synergy, we used RNA-sequencing data generated from specific sub-populations of hematopoietic cells to create gene lists that are relatively more (or less) expressed in either hematopoietic stem cells (HSCs) or leukemic stem cells (LSCs) compared to more differentiated populations, such as monocytes, lymphocytes, and all fully differentiated blood cells (Methods) [44]. Considering six pairs of cell types (Fig. 2.5f, left) leads to 12 gene lists; six of the genes lists represent more stem-like expression states, while the other six lists represent more differentiated signatures (Methods section for more details). For each gene list, we measured the correlation between the average expression of the genes in the list and drug synergy for each drug pair (Fig. 2.5f, right), and plotted correlations that were significant after multiple hypothesis testing correction.

Remarkably, we found two distinct sets of drug combinations – combinations that were more synergistic when applied to tumor samples with more stem-like expression profiles, and combinations that were more synergistic when applied to tumor samples with more differentiated expression profiles (Fig. 2.5f, right). For instance, many combinations containing the BCL-2 inhibitor Venetoclax were associated with increased synergy when a more differentiated signature was present. Specifically, these were most strongly associated with a monocytic expression signature (Signature D1). Recent studies have demonstrated that in some patients, AML subclones with a monocytic differentiation signature exist next to subclones with a more primitive, stem-like transcriptional profile [214, 149]. Monocytic subclones have been shown to be relatively resistant to Venetoclax [214, 149], raising the possibility that the drugs paired with Venetoclax in the identified combinations could be helping to overcome this resistance. For example, our approach identifies the combination of Ruxolitinib, a JAK inhibitor, with Venetoclax as having more synergy in more differentiated cancers. The capacity of Ruxolitinib to synergize with Venetoclax, specifically by targeting and overcoming monocytic resistance, has recently been demonstrated in several studies [148, 149]. EXPRESS identifies a number of additional drugs that may be combined with Venetoclax to the same effect, including the p38

MAP kinase inhibitor Doramapimod, the tyrosine kinase inhibitors Quizartinib and Sorafenib, the cyclin-dependent kinase 4/6 inhibitor Palbociclib, and the BET bromodomain inhibitor JQ-1. Interestingly, EXPRESS also identifies a handful of combinations not containing Venetoclax for which synergy is also associated with a differentiation signature, including the combination of Cabozantinib and Ruxolitinib, as well as the combination of MDM2 inhibitor Nutlin-3 and the chemotherapeutic cytosine analogue Cytarabine. To verify the importance of hematopoietic differentiation for AML drug sensitivity, we analysed the significance of these differentiation signatures in an additional, external dataset (Fig. 2.16). Using the AML cell line expression data and experimentally measured genetic dependency from the DepMap database, we found a significant association between the cancer cell line dependency of the genetic targets of the drug combinations in Fig. 2.5 and the expression of our differentiation signatures (empirical p-values 0.001, 0.001, and 0.026 according to three separate null models).

These results show that the exact position of AML cells on a hematopoietic differentiation spectrum predicts the synergy that can be achieved with specific therapy combinations. Assessment of an AML stemness (or differentiation) signature may therefore be useful in guiding therapy choices in the clinic.

2.2.6 Feature interactions identify drug-specific gene expression signatures

In addition to identifying expression signatures that are generally relevant for drug synergy across many combinations, our approach is also able to identify genes and pathways that are relevant for *specific* drugs. To quantify these drug-specific mechanisms, we used an extension of the Shapley value called the Shapley interaction index [87, 177], which extends attributions for single features to interactions between pairs of features (see Methods). Intuitively, expression of a particular gene may be more important when one of the drugs in a combination is specifically targeting that gene. Likewise, expression of a particular gene may be less important when neither drug targets that gene. Therefore, to quantify which genes were important for specific drugs, we measured the interaction values between each drug feature label and all gene features.

By analyzing the most important genes for each drug ranked by the average magnitude of their interactions (see Methods), EXPRESS is capable of revealing the specific biological processes related to synergy for a particular drug. After generating interaction values between all genes and drugs, we tested each list of global drug-specific gene attributions for pathway enrichment (see Methods). We found that these enrichments aligned with prior knowledge of the mechanisms of the drugs in question (Fig. 2.6).

For instance, EXPRESS pinpoints genes involved in apoptosis as important determinants of synergy for pairs of drugs containing Venetoclax (Fig. 2.6a, right), a drug which functions by restoring apoptotic function in malignant cells via inhibition of the gene B Cell Lymphoma-

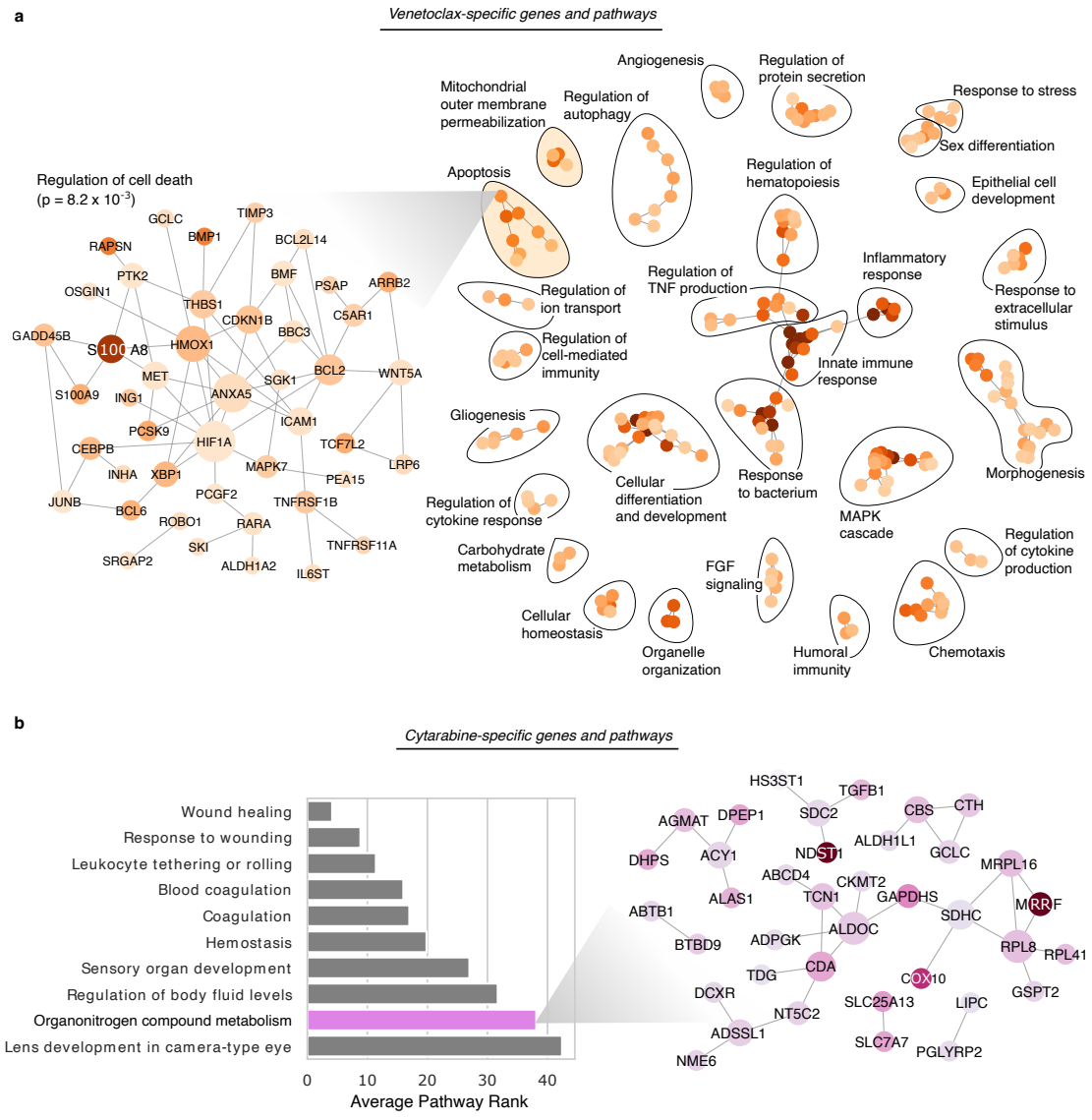

Figure 2.6: **Transcriptomic factors affecting synergy of combinations including specific drugs. a**, The top pathway enrichments in the set of transcripts affecting synergy of drug combinations including the drug Venetoclax. Each node in the graph on the right represents a single pathway, where the color indicates the strength of the enrichment, while edges indicate significant overlap in terms of the set of genes in each pathway. The zoomed inset graph on the left shows the genes in one pathway, "Regulation of cell death," from the cluster of apoptosis-related pathways. In the left inset graph, each node is a gene, while the edges represent known protein-protein interactions. **b**, The top pathway enrichments in the set of transcripts affecting synergy of drug combinations including the drug Cytarabine. The bar plot (left) shows the top pathways, while the inset graph on the right shows the relevant genes from one pathway, "Organonitrogen compound metabolism."

2 (BCL-2) [217]. Examining the individual genes in one of the enriched pathway modules for Venetoclax (Fig. 2.6a left, "Regulation of cell death" term, FDR-corrected $p = 8.0 \times 10^{-3}$) reveals Venetoclax's specific target BCL-2 to be an important predictive gene. Measuring the strength and direction of the relationship between BCL-2 expression and Venetoclax-specific BCL2 attribution values, we find that increased BCL2 expression is associated with markedly increased drug synergy in the context of Venetoclax treatment (Spearman $\rho = 0.156$, $p = 4.0 \times 10^{-70}$). Other genes in this module include S100A8 and S100A9, both genes which have previously been linked to patient response to Venetoclax as well as differential expression in hematopoietic stem cells compared to more differentiated populations [122, 154]. Other important biological processes detected by the Venetoclax-specific attributions include the MAPK cascade, which has been linked to Venetoclax-resistance through stabilization of MCL1 [96], and Fibroblast growth factor (FGF) signaling (see Fig 2.6a, right). Interestingly, Fibroblast Growth Factor 2 (FGF2) release by dying cells has recently been implicated as a transient, non-heritable mechanism of Venetoclax resistance [22], highlighting the power of transcriptomic analysis to discover phenomena not observable in mutational data alone.

As another example, one of the most enriched biological process terms for cytarabine, an organonitrogen compound, is the "metabolism of organonitrogen compounds" (Fig. 2.6b, $p = 4.0 \times 10^{-5}$ from hypergeometric test, FDR-correction using Benjamini-Hochberg procedure). The individual genes in this module include CDA and NT5C2, two genes responsible for the metabolism of cytarabine that have previously been shown to be important genetic factors determining the response to cytarabine therapy [153]. We conducted the interaction drug-specific feature attribution analysis and pathway enrichment characterization for all drugs, which can serve as a resource for researchers interested in the particular mechanisms underlying AML response to these drugs. This analysis demonstrates that EXPRESS is able to identify not only expression trends important for large sets of combinations, but also for specific drugs.

2.3 Discussion

By ensembling complex models, the EXPRESS framework not only enables accurate predictive performance, but also robust and biologically meaningful explanations. While prior work has been able to attain high accuracy with complex models [220], our approach can provide explanations to assure patients, clinicians and scientists of the biological soundness of our predictions, even when models have high-dimensional input features with a high degree of feature correlation. The importance of interpretability in the context of biomedical AI is increasingly being recognized. Model explanations can help identify when apparently accurate "black box" models may in fact be relying upon unreliable confounders, also known as "shortcuts" [52, 79]. Explanations also allow physicians to communicate the logic of

algorithmic decisions with patients, which can increase patient trust in the treatment process [145]. Finally, by displaying the logic underlying model decisions, explainable AI can enable better collaboration between physicians and AI models. For example, when applied to the Beat AML dataset [286], our model was optimized without respect to the cost or FDA approval status of different drug combinations. Where a "black box" model can only provide physicians with a synergy score for drug combinations, the mechanistic explanations provided by our model could help a physician to choose combinations with a similar predicted mechanism that might be preferable in terms of cost or FDA approval status.

As the application of explainable AI in the life sciences continues to grow, we anticipate that our framework will be broadly helpful to researchers. As observed in previous work, model prediction and model explanation are not always identical tasks [34, 62], and understanding how to create approaches that work for both of these goals is important given the popularity of Shapley value-based explanations for complex models. By demonstrating the high degree of variability in explanations within a class of models (Fig. 2.2-2.3), we hope to discourage users from naively selecting a single model to explain, and instead encourage users to explain ensembles of models. While our work focused on transcriptomic data, the high degree of feature correlation and dimensionality is also characteristic of many other forms of 'omics data, indicating the broad impact of these results. We envision that future work on more efficient approaches to create ensemble models, which can be computationally costly, will be valuable. Likewise, further theoretical characterization of the feature attributions of complex models, such as deep neural networks and gradient boosting machines, will likely be important. While recent work has theoretically characterized the heterogeneity in feature importance across different well-performing models from the same model class, this work has thus far been limited to a small number of simple model classes (linear regression, logistic regression, and simple decision trees) [60].

In parallel to this work on improving the quality of attributions for black-box models, another thread of contemporary research focuses on incorporating prior biological knowledge into the modeling process. This includes methods like MERGE, which regularizes the coefficients of linear models using multiomic prior information [157], as well as Attribution Priors, which uses an efficient and axiomatic feature attribution method to align deep neural network attributions with biological priors during the training process [65, 295]. Other methods to incorporate biological prior information focus on structurally modifying neural network architectures, limiting interaction to genes that are known to share biological processes [143, 93]. Determining the best way to attribute feature importance in the context of the structurally-modified models will be important future work. Similarly, understanding how explainable AI can be optimally combined with the *unsupervised* deep learning models that have been successful in the context of single cell gene expression data will be another important line of future work [171, 58].

When applied to a large dataset of *ex vivo* drug synergy measurements in primary tumor cells from patients with AML, EXPRESS can both accurately predict drug synergy, as well as uncover a differentiation-related expression signature underlying the predictions for many combinations. While mutational status is increasingly considered in the clinical management of AML, our study demonstrates how useful tumor expression data can be for the prediction of drug combination synergy. Our experiments show that the extent of hematopoietic differentiation of AML cells is an important factor for the prediction of the synergy that can be achieved with specific therapy combinations, which has potential clinical application. One limitation of the current study is that our approach was applied to a dataset of drug synergy measurements in bulk tumor samples, rather than synergy assayed in specific purified tumor cell populations. As more studies come out measuring the specific effects of anti-cancer drugs on the heterogeneous inidividual cells and subpopulations comprising AML [149, 214], applying EXPRESS to these datasets may yield interesting additional mechanistic insights.

2.4 Methods

2.4.1 Feature attribution methods

Shapley values

The Shapley value is a concept from coalitional game theory designed to fairly distribute the total surplus or reward attained by a coalition of players to each player in that coalition [252]. For an arbitrary coalitional game, $v(S) : \mathcal{P}(S) \mapsto \mathbb{R}$ (where S is the set of players and $\mathcal{P}$ indicates the powerset), the Shapley value for a player i is defined as the marginal contribution of that player averaged over the set of all $d!$ possible orderings R of the d players in S:

$$\phi(i) = \frac{1}{d!} \sum_{R} v(S_i^R \cup i) - v(S_i^R), \tag{2.1}$$

where S_i^R indicates the set of players in S preceding player i in order R.

To use this value to allocate credit to features in a machine learning model, the model must first be defined as a coalitional game. Deciding exactly how to define a model as a game is non-trivial, and a variety of different approaches have been suggested [267, 179, 45, 37]. The most popular, SHAP [179], defines the game as the conditional expectation of the output of a model f for a particular input sample $x \in \mathbb{R}^d$ given that the features in S have been observed:

$$v(S) = \mathbb{E}[f(x)|x_S]. \tag{2.2}$$

Because modeling an exponential number of arbitrary conditional distributions is often intractable, in practice the simplifying assumption that input features are independent is often made, allowing the expected value to be calculated over the marginal distributions of the features not in each given set, rather than the conditional distributions [179].

In our benchmark experiments, because comparable attributions are desirable for both the gradient boosted machine and neural network models, and because we want *global* attributions (features which are important for across all samples in the dataset), we use the SAGE software package to generate attributions. SAGE values define the coalitional game as the average reduction in test error $\ell(\cdot, \cdot)$ when a set of features are included as compared to the base rate prediction $f_\emptyset(X_\emptyset)$:

$$v(S) = \mathbb{E}[\ell(f_\emptyset(X_\emptyset), Y)] - \mathbb{E}[\ell(f(X_S), Y)]. \tag{2.3}$$

Since the SAGE package uses a sampling approach over possible coalitions of features to estimate Shapley values, it is important to ensure that the estimates are well-converged. To ensure convergence for the synthetic benchmark experiments, 102,400 permutations were used for all experiments (see Fig. 2.17).

For experiments using the full BEAT AML dataset, we explained models using TreeSHAP [177]. TreeSHAP is a model-specific algorithm that leverages the structure of tree-based machine learning models (like XGBoost, the best performing model class for the problem) to quickly calculate SHAP values in polynomial time. TreeSHAP tries to approximate the conditional expectations using the conditional distribution defined by the tree structure. In instances where we needed global TreeSHAP attributions, we follow Lundberg et al. [177] and define the global attribution as the average magnitude of the local explanations ϕ_i over the whole dataset $\mathcal{D}$:

$$\Phi_i(f, \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}} |\phi_i(f, x)|. \tag{2.4}$$

In instances where we wanted global attributions that were also directional, we considered the correlation between the SHAP attributions for a feature and that feature's underlying value:

$$\rho_{X_i, \phi_i} = \frac{\text{cov}(X_i, \phi_i)}{\sigma_{X_i} \sigma_{\phi_i}}. \tag{2.5}$$

Other attributions

In addition to Shapley values, we also considered five other feature attribution methods in various experiments. Implementations of DeepLift and Integrated Gradients were from the Captum library [134], while implementations of Gain and Cover were default feature importance methods in the XGBoost library. Another model agnostic method, LIME [230],

was considered, but ultimately could not be used because of computational efficiency problems. For example, explaining even a single sample from the Beat AML dataset (which consists of 12,362 samples samples) took over 30 minutes with LIME. In contrast, explaining *12,362 samples* (each having 15,535 features) of the same model with TreeSHAP took 5.62 seconds on our CPU server (96 CPUs).

Models used in benchmarks

In our benchmark tests, we evaluated two complex model classes explained using Shapley values. The first model class consisted of feed-forward neural networks. To train these networks, we used the PyTorch deep learning library [211]. To tune the models, we did a grid search across the following parameters: we used between 2 and 4 fully connected layers with either 'ELU' or 'ReLU' activations; we used a number either 64, 128, or 256 nodes in the first hidden layer and considered both a 'decreasing' and a 'non-decreasing' architecture (where 'decreasing' reduced the number of nodes in each successive layer by a factor of 2, and non-decreasing maintained a constant number of nodes across layers). We then trained the networks using the Adam optimizer with a learning rate of 0.001 for a maximum of 1,000 epochs. Early stopping was used to stop the training process if the mean squared error loss did not improve after 50 epochs. The second model class consisted of gradient boosted machines. To train these moodels, we used the XGBoost library [39]. To tune the models, we again did a grid search across several parameters: we considered a max tree depth of either 2, 10, 18, 26, 34, or 42; we also considered a range of 'eta' parameters including either 0.3, 0.2, 0.1, 0.05, 0.01, or 0.005. All models were boosted for 1000 rounds, and the saved model with best validation error was used for downstream prediction and explanation.

In addition to Shapley values applied to neural networks and gradient boosted machines, we also compared to a baseline of three more classical feature attribution methods used in biological feature discovery. The first involves ranking features X according to their Pearson correlation ρ with the outcome of interest Y:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}. \tag{2.6}$$

Ranking features in this way can be viewed as a special case of a family of feature selection algorithms known as Backward Elimination with the Hilbert-Schmidt independence criterion (BAHSIC) [264]. We also ranked features according to the magnitude of their coefficients in an elastic net regression, which is a linear regression where both the $\ell 1$ and $\ell 2$ norm of the coefficient vector are penalized in the loss function [315]. To train elastic net regression models, we used the ElasticNetCV function in the scikit-learn library with number of folds set to 5 [213]. Finally, we also tested a procedure known as recursive feature elimination

using support vector machines (SVM RFE) [94]. As an estimator for this algorithm, we used the epsilon-Support Vector Regression function in scikit-learn with a linear kernel, then used the RFE function from the same library to select features with the parameters 'n_features_to_select' and 'step' set to 1.

2.4.2 Benchmark evaluation metric

To evaluate how well different approaches recover biologically-relevant signal, we designed a simple benchmark metric to evaluate the concordance between a list of features ranked by machine learning approaches and a ground truth list of features. It was necessary to design a new benchmark metric because existing metrics tend to evaluate how well feature attributions identify the features that are important for a particular machine learning model [177]. Our feature discovery benchmark measures how well each approach recovers biological signal by plotting the number of *true features* cumulatively found at each point in the list of features ranked by that approach, then summarizing this curve by measuring the area beneath it using the 'auc' function in scikit-learn [213] (see Fig. 2.8). A larger area under the feature discovery curve (AUFDC) corresponds to better performance. A perfect score for a model with 10 true features out of 100 true features would be 950, while a random ordering would be expected to achieve an AUFDC of 500 on average. In order to make this score more intuitive, we subtract the random score of 500 and divide by the maximum possible area greater than random (450) so that the scores are scaled between 0 and 1, where 0 now means random performance and 1 means perfect performance.

2.4.3 Synthetic Datasets

In order to use our benchmark evaluation metric to determine how well different approaches could uncover underlying biological signal, it was essential to define datasets where the ground truth is known. Creating synthetic datasets also gave us the direct control needed to gain deeper understanding into the factors impacting the success of these algorithms, such as feature correlation, noise, and outcome type. We tested feature discovery performance on 240 total synthetic or semi-synthetic datasets. Each dataset comprised a feature matrix $X \in \mathbb{R}^{n \times d}$, where n represented the number of samples and d represented 100 input features, and an outcome vector $y \in \mathbb{R}^n$ which is some function of the original features ($y = f(X)$).

We considered three groups of distributions for the feature matrices. The first group was 1000 samples of 100 independent Gaussian features randomly generated to be 0 mean, unit variance. The second group was 1000 samples of 100 Gaussian features with ten groups of five tightly correlated features (Pearson's $\rho = 0.99$). The final group involved 223 real patient gene expression samples from the Beat AML Dataset [286].

We considered four different functions f by which the features X were related to the outcome y. The first function was a linear function with 10 non-zero coefficients, $f(X) = X\beta$. The second function was the sum of 10 univariate ReLU functions $f(X) = \sum_{i=0}^{10} \text{ReLU}(x_i)$. The third function was a sum of 10 pairwise multiplicative interactions $f(X) = \sum_{i=0}^{10} x_i x_{i+1}$. The final function was a sum of 10 pairwise AND functions $f(X) = \sum_{i=0}^{10} \left(x_i > 0 \wedge x_{i+1} < 0\right)$. For each of the twelve possible pairwise combinations of feature matrices and outcome functions, we created 20 specific datasets meeting the specifications, where the only difference was that the features were randomly regenerated (or randomly re-sampled from the full transcriptome in the case of the AML features), and the features selected as true features were re-selected.

2.4.4 Comparing ensembles and individual models

To train ensemble models for comparison in our benchmark experiments, we used the method of bootstrap aggregation, or *bagging* [25]. This method involves first bootstrapping the data, or resampling the dataset with replacement until the bootstrapped dataset has as many samples as the original, then training a model on the bootstrap resampled dataset. We repeat the process of bootstrapping and training models 20 times. Since our benchmark is a regression problem, the 20 model outputs are then aggregated by a simple mean. This method is known to improve predictors by increasing their stability.

To understand the difference in quality of the individual model attributions and the ensemble model attributions, we considered two separate objective metrics. The first was to assess the *stability* of the attributions. We measured the pairwise cosine similarity of 20 ensemble models' attributions trained on bootstrap resampled versions of each dataset, then measured the pairwise cosine similarity of 20 individual models' attributions trained on bootstrap resampled versions of the same datasets:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2}. \tag{2.7}$$

The next metric aimed to understand how much importance was put on truly important features compared to how much was potentially placed on spurious correlates. We therefore measured the Gini index of each global attribution vector to understand how *sparse* of an attribution was learned by each model:

$$G(x) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}} \tag{2.8}$$

2.4.5 Beat AML Datasets

The Beat AML program comprises a large cohort of AML patient tumor samples for which *ex vivo* anti-cancer drug sensitivity has been measured. Since our project aimed to uncover the transcriptomic factors underlying anti-cancer drug synergy, we only included patients from the cohort whose tumors had been characterized with RNA sequencing, for which measurements pairs of anti-cancer drugs had been tested. Our final dataset contained the RNA-sequencing expression data from 285 patients with myeloid malignancy, and drug synergy measured on a subset of patients for 131 combinations of 46 distinct drugs.

The input features used in modeling each of 12,362 samples (where a sample is one patient and one combination of two anti-cancer drugs) were represented as a vector $x \in \mathbb{R}^{15535}$. This vector is constructed by concatenating three other vectors. First, we describe each patient's tumor sample using a vector of gene expression values (RNA-seq data – see RNA-seq pre-processing section for more information), $g \in \mathbb{R}^{15377}$. We described each drug combination using a feature vector, $v \in \mathbb{R}^{46}$, of drug identity labels where each element v_i was equal to 1 if the ith drug was present in the combination and 0 otherwise. We also incorporated drug target information for each drug combination, using information compiled from DrugBank plus a supplementary literature search for reliable drug targets, for a total set of 146 targets.1 We then described the drug targets of each combo with a vector $u \in \mathbb{R}^{146}$, where each element u_j was equal to 2 if the jth target was targeted by both drugs, equal to 1 if the jth target was targeted by only one of the drugs, and equal to 0 if the jth target was not targeted by either drug.

2.4.6 RNA-seq preprocessing

To ensure a quality signal for prediction while removing noise and batch effects, it is necessary to carefully preprocess the RNA-seq gene expression data. In this study, the RNA-seq were preprocessed as follows. First, raw transcript counts were converted to fragments per kilobase of exon model per million mapped reads (FPKM). FPKM is a more reflective of the molar amount of a transcript in the original sample than raw counts, as it normalizes the counts for different RNA lengths and for the total number of reads. FPKM is calculated as follows:

$$\text{FPKM} = \frac{X_i \times 10^9}{N l_i}, \tag{2.9}$$

where X_i represents the raw counts for a transcript, l_i is the effective length of the transcript, and N is the total number of counts.

After converting counts to FPKM, we removed any non-protein-coding transcripts from the dataset. We also removed transcripts that were not meaningfully observed in our dataset by dropping any transcript where $> 70\%$ measurements across all samples were equal to 0.

We then log-transformed the data and standardized each transcript across all samples, such that the mean for that transcript was equal to zero and the variance of the transcript was equal to one. Finally, we corrected for batch effects in the measurements using the ComBat tool available in the sva R package [158].

2.4.7 Drug synergy metric

The outcome in our model was drug synergy: whether a number of drugs exhibit more anti-cancer activity in combination than would be expected simply by adding their individual activities together. We therefore calculated synergy using the Combination Index (CI) of the two drugs:

$$\mathrm{CI} = \frac{\mathrm{IC50}_1^{\mathrm{combination}}}{\mathrm{IC50}_1^{\mathrm{single}}} + \frac{\mathrm{IC50}_2^{\mathrm{combination}}}{\mathrm{IC50}_2^{\mathrm{single}}} \tag{2.10}$$

where $\mathrm{IC50}_i^{\mathrm{single}}$ is the dose of drug i required to reduce cell viability to 50% when used alone and $\mathrm{IC50}_i^{\mathrm{combination}}$ is the dose of drug i required to reduce cell viability to 50% when used in combination with the other drug we are measuring [41]. When a drug combination is synergistic, the CI will be less than 1 (it will be equal to 1 when the combination is additive and greater than 1 when the combination is antagonistic). In our model, we log-transformed the CI measure to help manage the skewness of the original distribution, and then scaled the measure to make the distribution 0 mean and unit variance. We also multiplied by -1 for ease of interpretation: more synergistic combinations thus have a larger score.

While prior studies have made use of response-surface analysis, which involves measuring the volume between an idealized additive response surface and a measured actual response surface (9,10), these measures could not be applied to the “diagonal” measurements present in the Beat AML Dataset. A major drawback to response-surface analyses is they requires a "checkerboard" of measurements at different drug concentrations, where the ratios and doses of each drug in a combination is varied. This consumes many more cancer cells, which is problematic when using primary cells from patients, as the amount of sample that can be collected is more limited than when using cell lines.

2.4.8 Cross-validation and sample stratification

In addition to the model parameters which are learned from data, machine learning models also rely on hyperparameters, which must be tuned to a specific task in question in order to attain optimal predictive performance. In order to estimate the true generalization error of a model (i.e., how well that model is likely to perform on unseen data), it is essential that model parameters and hyperparameters must be learned and chosen based on training

data, while predictive performance is evaluated on a held-out test set that is never used for hyperparameter selection or model training. Hyperparameters are typically picked through a cross-validation (CV) procedure which determines the optimal hyperparameters for the model by validating them through a number of internal training and validation fold pairs randomly chosen from the set of training samples used for learning the model parameters.

To effectively train our models and evaluate predictive performance, we therefore utilized a nested 5-fold CV procedure, whereby the data were split into 5 separate test folds. For each of these test folds, we trained our synergy prediction model using the four remaining folds and evaluated it on the held-out test fold. To properly tune the hyperparameters of the models trained for each test fold, three of the four training folds were used as an internal training set, while the remaining fold was used as a validation set. The hyperparameters were selected by an inner loop, where for each hyperparameter set of interest, the model was trained on the internal training set and tested on the validation set. The hyperparameters giving the best performance on the validation set were then used to train a model on the entire training data, which was then finally evaluated on the held-out test fold. The grid of hyperparameters tested for each model type are as follows. For the sklearn elastic net implementation, the "alpha" parameter was tuned over values ranging from 0.1 to 100, while the "l1_ratio" parameter was tuned from 0.25 to 0.75. For the sklearn Random Forest implementation, the "n_estimators" parameter was tuned from 128 trees to 2048 trees, while the "max_features" parameter was set to be either "log2," "sqrt," or "256." For XGBoost, "max_depth" was tuned between 4 and 8, "subsample" was tuned to values between 0.1 and 0.8, and learning rate was tuned between 0.05 and 0.1. For deep neural networks, hyperparameters were tuned following the grid given in Preuer et al. [220], where an additional data pre-processing step that would optionally transform the RNA-seq features with a hyperbolic tangent function in addition to standardization was also included as a hyperparameter. Code for tuning these networks was found at `https://github.com/KristinaPreuer/DeepSynergy`. For both deep neural networks and gradient boosted machines, early stopping based on validation set error was used to choose the number of epochs/estimators.

In order to evaluate the model's performance for a variety of hypothetical uses, we stratified our data into training and testing sets in four different ways (Fig. 2.4). Each sample in our dataset consists of a synergy measurement for a 2-drug combination tested in a patient's tumor cells. In the first stratification setting, we ensured that any sample (2-drug combination and patient) present in the test data would never be present in the training data. The second setting maintains the first setting's requirement that each sample in the test data be novel, but additionally ensures that any combination of drugs in the test data would never be present in the training data. The third setting maintains the first setting's requirement that each sample in the test data be novel, but additionally ensures that any patient in the test data would never be present in the training data. Finally, the fourth setting maintains the

first and second settings' requirements, while additionally ensuring that for any combination of drugs in the test data, at least one of the drugs in that combination would never have been present in the training data. Each of these settings should be increasingly difficult to predict, as each setting requires progressively more generalizable trends in the data to have been learned.

2.4.9 XGBoost model ensembles

After selecting XGBoost as the best-performing model class for the prediction of anti-AML drug synergy, we then wanted to account for the full diversity of possible good XGBoost models fit to the highly correlated AML gene expression data. We therefore trained 100 models and explained the ensemble model. Each individual model had both row and column subsampling turned on for each additional tree fit, and the difference between the models in the ensemble was the random seed given to generate the subsampling.

In practice, instead of explaining the entire ensemble (the average output of each of the 100 models), we instead explain each individual model and average the explanations. This is possible due to the linearity property of Shapley values [199]. This property states that for the convex combination of any two coalitional games v and w, the attribution for player i will be the convex combination of the attributions that player would attain in each individual game v and w:

$$\phi_i(\alpha v + (1-\alpha)w) = \alpha\phi_i(v) + (1-\alpha)\phi_i(w). \tag{2.11}$$

2.4.10 Overall pathway analysis

The highest-ranked genes in the lists ordered by global Shapley values were tested for pathway enrichments using the StringDB package in R [277]. The package was initialized using the arguments: 'version' = '10', 'species' = '9606', and 'score_threshold' set to the default of 400. We used the set of pathways from KEGG (Kyoto Encyclopedia of Genes and Genomes) for enrichment tests. The actual enrichments were calculated by a hypergeometric test implemented in the 'get_enrichment' method. In order to ensure that pathway enrichments were robust to the threshold used for selecting the highest-ranked genes, we averaged the enrichment test result over a variety of different thresholds, ranging from 200 to 800 top genes. FDR correction was applied using the Benjamini-Hochberg procedure [17].

2.4.11 Generation of Differential Expression Stemness Profiles and Measurement of Synergy Correlation

To generate expression signatures related to more or less-differentiated states of cells in the hematopoietic cell lineage, we downloaded RNA-seq data from isolated cells from particular levels of developmental transitions [44]. We then used the R package DESeq2, which tests for differential expression in RNA-seq data based on a negative binomial model, to generate lists of genes upregulated in particular populations of cells as compared to other populations [175]. The populations we compared were as follows: monocytes vs. hematopoietic stem cells (HSCs), lymphocytes vs. HSCs, all fully differentiated cells (which included erythroblasts, T cells, B cells, NK cells, and monocytes vs. HSCs, monocytes vs. leukemic stem cells (LSCs), lymphocytes vs. LSCs, and all fully differentiated cells vs. LSCs. The immunophenotypes used to sort HSCs and LSCs were Lin– CD34+ CD38– CD90+ CD10– and Lin– CD34+ CD38– TIM3+ CD99+, respectively. Gene expression profiles for these populations were the same used in [44]. The multiple testing-adjusted p-value used as a significance threshold for differentially-regulated genes was 0.05.

When we tested for association between our differential expression profiles and synergy for particular combinations of drugs, we first considered only samples containing the drug combination in question. We then averaged gene expression over all genes in the differential expression profile. Finally, we measure the Pearson correlation between the average expression profile and the drug combination synergy for those samples. Since we have many combinations of drugs and many differential expression profiles to test, we correct for multiple testing using the Benjamini-Hochberg FDR correction procedure [17]. We then display only correlations that are significant after correction. Additionally, we only want to consider correlations that are robust to the differences in the particular stemness-differentiation profiles, so we only plot correlations for drugs that are significant across at least two profiles.

2.4.12 Cancer Dependency Map Analysis

To externally validate the importance of the hematopoietic differentiation expression signature for the drug combinations identified in Fig. 2.5f, we used data from the Cancer Dependency Map (DepMap) database. Specifically, we downloaded the Genetic Dependency CRISPR assays (DepMap 21Q4 Public+Score, Chronos, 'CRISPR_gene_effect.csv') and the expression data (21Q4 Public, 'CCLE_expression.csv'), as well as the metadata in the Cell Line Sample Info file ('sample_info.csv'), from the DepMap portal. After downloading this data, we filtered it so that only cell lines with the lineage subtype 'AML' were present in the analysis. Then, we test for association between the average expression of the signature with the most associated drug combinations from Fig. 5f (S1/D1) and the genetic dependency of the targets of the drug combinations listed in Fig. 2.5f.

In order to assess the significance of the associations, we design three null models. First, we generate a null distribution by randomizing the genes that are averaged to calculate the expression signature 1000 times, and measuring average magnitude of the Spearman correlation of these random signatures with the genetic dependency Chronos scores of the genetic targets of the targets of the drug combinations listed in Fig. 2.5f ("Randomize Pathway Genes" null). Second, we generate a null distribution by randomly permuting the rows (cell lines) in the gene expression matrix 1000 times, before measuring the average magnitude of the Spearman correlation of the average expression of the hematopoetic differentiation expression signature with the genetic dependency Chronos scores of the genetic targets of the drug combinations listed in Fig. 2.5f ("Permute Pathway Expression" null). Third, we generate a null distribution by leaving the expression matrix unpermuted and unrandomized, and measuring the average magnitude of the Spearman correlation of the average expression of the hematopoetic differentiation expression signature with the genetic dependency Chronos scores of random sets of genes (where the random sets are constrained to be the same size as the number of true targets of the set of combinations in Fig. 2.5f, "Randomize Targets" null). Across all three null models, we find that the true expression signature is significantly associated with cancer cell line genetic dependency on the drug targets of the drugs in Fig. 2.5f (empirical p-values 0.001, 0.001, and 0.026 for the "Randomize Pathway Genes," "Permute Pathway Expression," and "Randomize Targets" nulls, respectively).

2.4.13 Drug-specific pathway analysis

To analyze the biological processes relevant for combinations containing specific drugs in the dataset, we tested the top-ranked genes in the lists ordered by the average magnitude Shapley interaction indices [87, 177]. Following the same procedure described above, we calculated pathway enrichments using the StringDB package in R [277]. We used the set of pathways from Gene Ontology (GO) Biological Process terms for enrichment tests. The enrichments were calculated by a hypergeometric test implemented in the 'get_enrichment' method. In order to ensure that pathway enrichments were robust to the threshold used for selecting the highest-ranked genes, we averaged the enrichment test results over a variety of different thresholds, ranging from 200 to 800 top genes. FDR correction was applied using the Benjamini-Hochberg procedure [17].

For Venetoclax, since a large number of biological process terms were signficicantly enriched, and since there is substantial overlap and similarity between these gene sets, we clustered the significantly enriched pathways into modules. We defined an adjacency matrix where each gene set represented a node in a network, and the Jaccard Index (a measure of overlap) between pathways was used to define edges. We binarized the matrix for pathways with Jaccard Index greater than 0.4. We then manually annotated all connected components in the resultant graph. To plot the network, we used the spring layout functionality in the

networkx library in Python [95].

2.5 Data availability

The results published here are in part based upon data generated by the Cancer Target Discovery and Development (CTD2) Network (`https://ocg.cancer.gov/programs/ctd2/data-portal`) established by the National Cancer Institute's Office of Cancer Genomics. Sequencing data are available in the GDC data portal under dbGaP Study Accession phs001657. The Beat AML patient sample data used in this study was done under an early access agreement, prior to final accrual, harmonization, and public release of the full dataset. As such, the subset of samples included in this study may differ in sample representation, quality control thresholds, and data normalizations from those found in GDC and in the final study describing the full dataset.

The code necessary to reproduce synthetic datasets can be found at `https://github.com/suinleelab/express`.

2.6 Code availability

Code necessary to reproduce our experimental findings can be found at `https://github.com/suinleelab/express`.

2.7 Extended Data

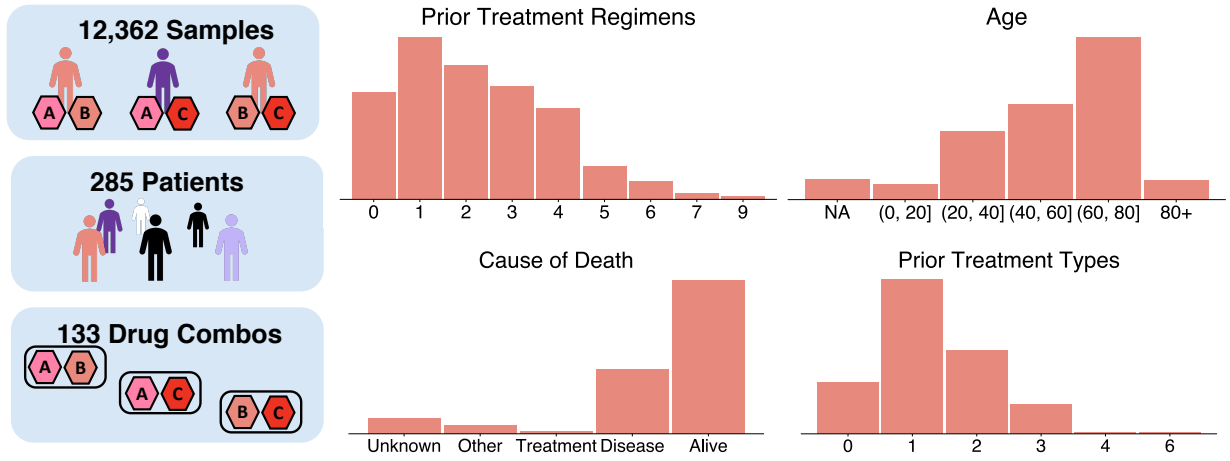

Figure 2.7: **Descriptive statistics of Beat AML cohort.** Histograms showing the relative density of prior treatment regimens, age, cause of death, and prior treatment types in the cohort of 285 patients in our dataset, which consisted of 12,362 samples with paired gene expression and drug synergy measurements for 133 pairs of 46 anti-cancer drugs.

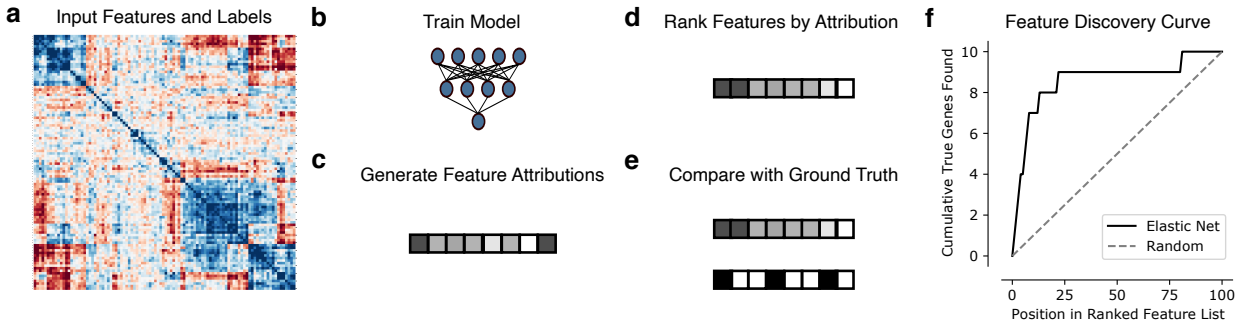

Figure 2.8: **Feature discovery benchmark.** For each synthetic or semi-synthetic dataset (**a**), we trained a variety of models (**b**) including neural networks, gradient boosted machines, support vector machines, and elastic net regression, as well as univariate statistics (Pearson correlation). For the machine learning models, we then used SAGE to generate global Shapley value feature attributions (**c**), ranked the features according to the magnitude of their attributions (**d**), and compared the ranked list generated by each method to the binary ground truth importance vector (**e**). To measure the feature discovery quality of each method, we plotted how many "true" features are found cumulatively at each point in the ranked feature list (**f**), then summarized the curve generated by this procedure by measuring the area under the feature discovery curve (AUFDC). This score is then rescaled so that a score of 0 represents random performance while a score of 1 represents perfect performance.

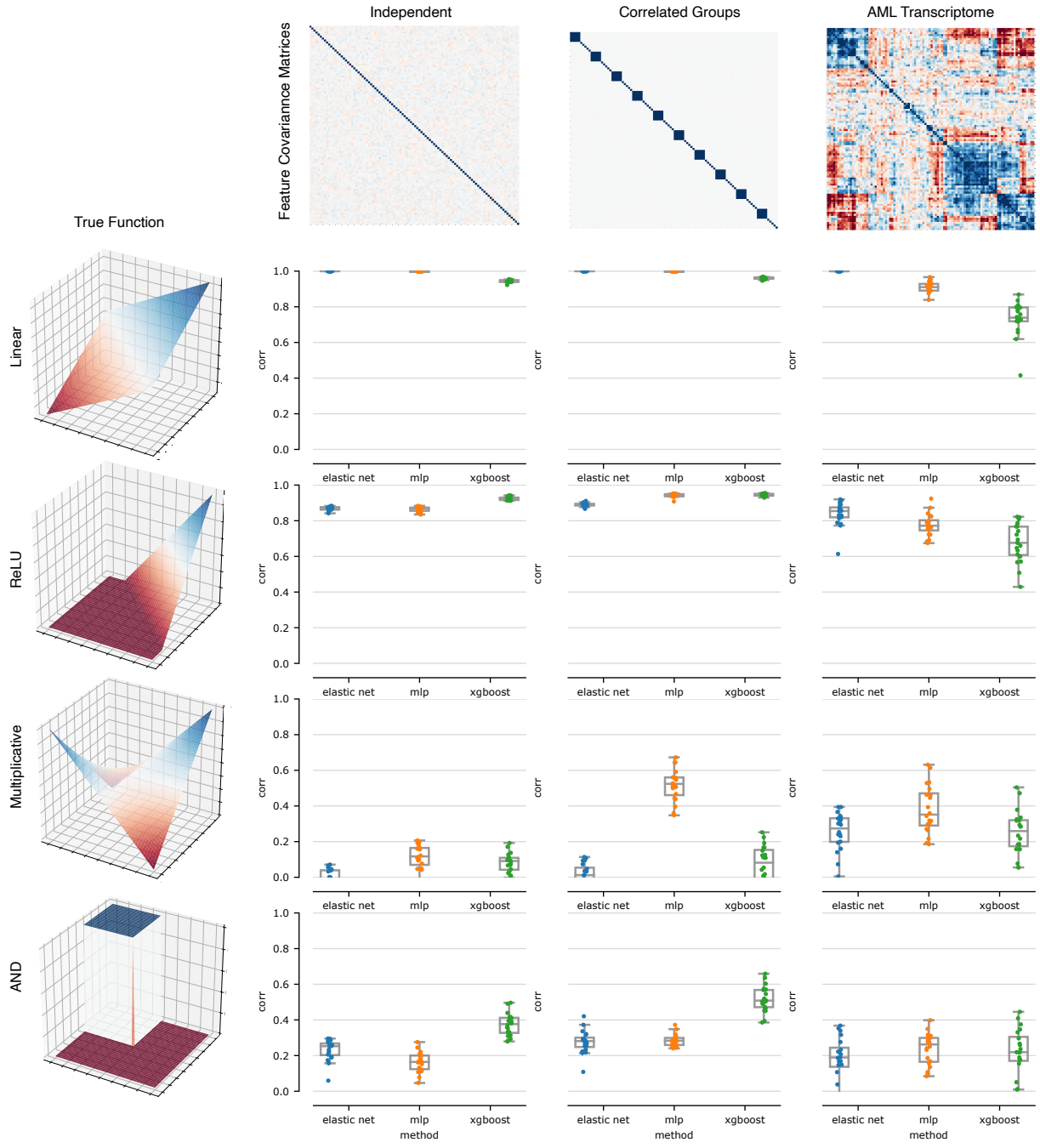

Figure 2.9: **Predictive performance of models trained synthetic datasets.** Predictive performance, as measured by the Pearson correlation of the predicted and true labels for the models trained in the benchmark presented in Fig. 2.2.

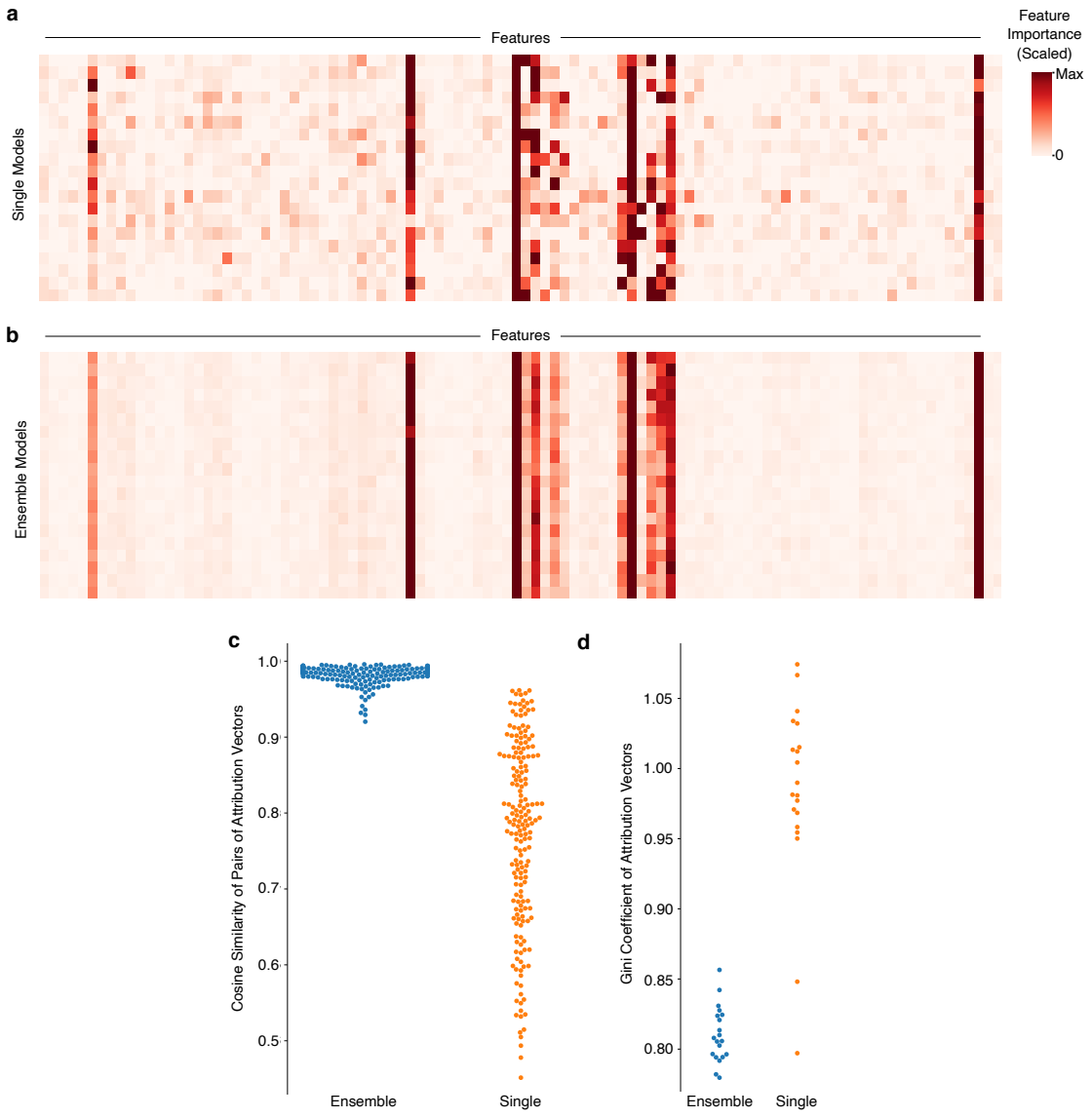

Figure 2.10: **Ensembling overcomes the variability in attributions present in individual models.** To understand why ensemble models were able to attain better feature discovery performance than single models, we compared the characteristics of the attribution vectors of XGBoost models trained on bootstrap resampled versions of a Correlated Groups dataset with a step-function outcome. **a**, Heatmap of feature attributions for 20 individual XGBoost models. **b**, Heatmap of feature attributions for 20 ensembles of XGBoost models. **c**, Pairs of attribution vectors from ensembled models are more similar across bootstrap resamples of the dataset than attribution vectors from single models, as measured by cosine similarity. **d**, Attribution vectors from ensembled models place a larger proportion of their importance on a smaller set of features than attribution vectors from single models, as measured by the Gini coefficient of the attribution vectors, a measure of vector sparseness.

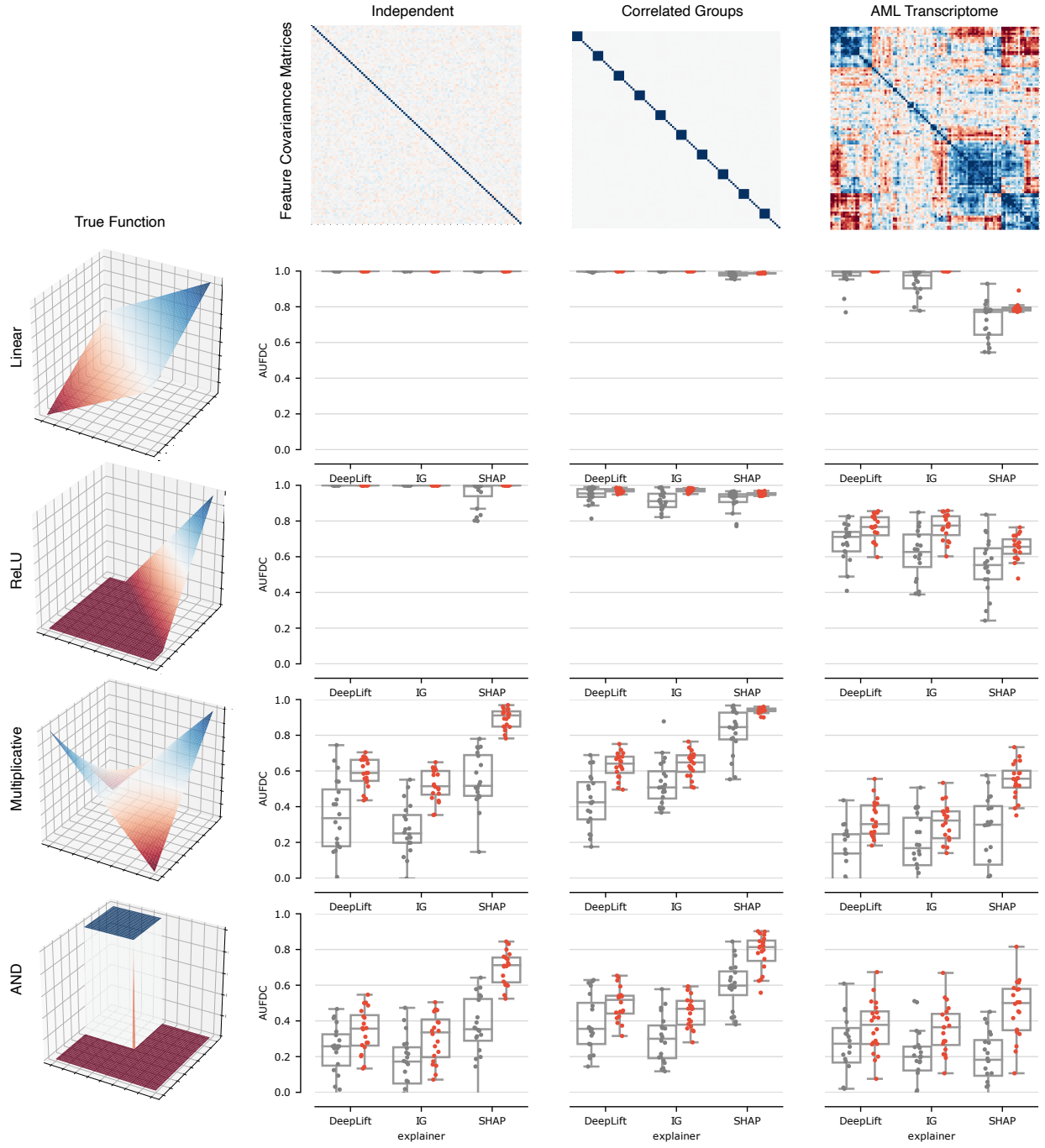

Figure 2.11: **EXPRESS improves feature attributions of deep learning models.** Comparison of feature discovery performance between individual deep learning models and ensembles of deep learning models across all 12 dataset types from the synthetic benchmark. Three separate feature attribution methods are tested for each model: a) DeepLift [256], b) Integrated Gradients [274], and c) SHAP (in this case implemented as SAGE) [45, 179].

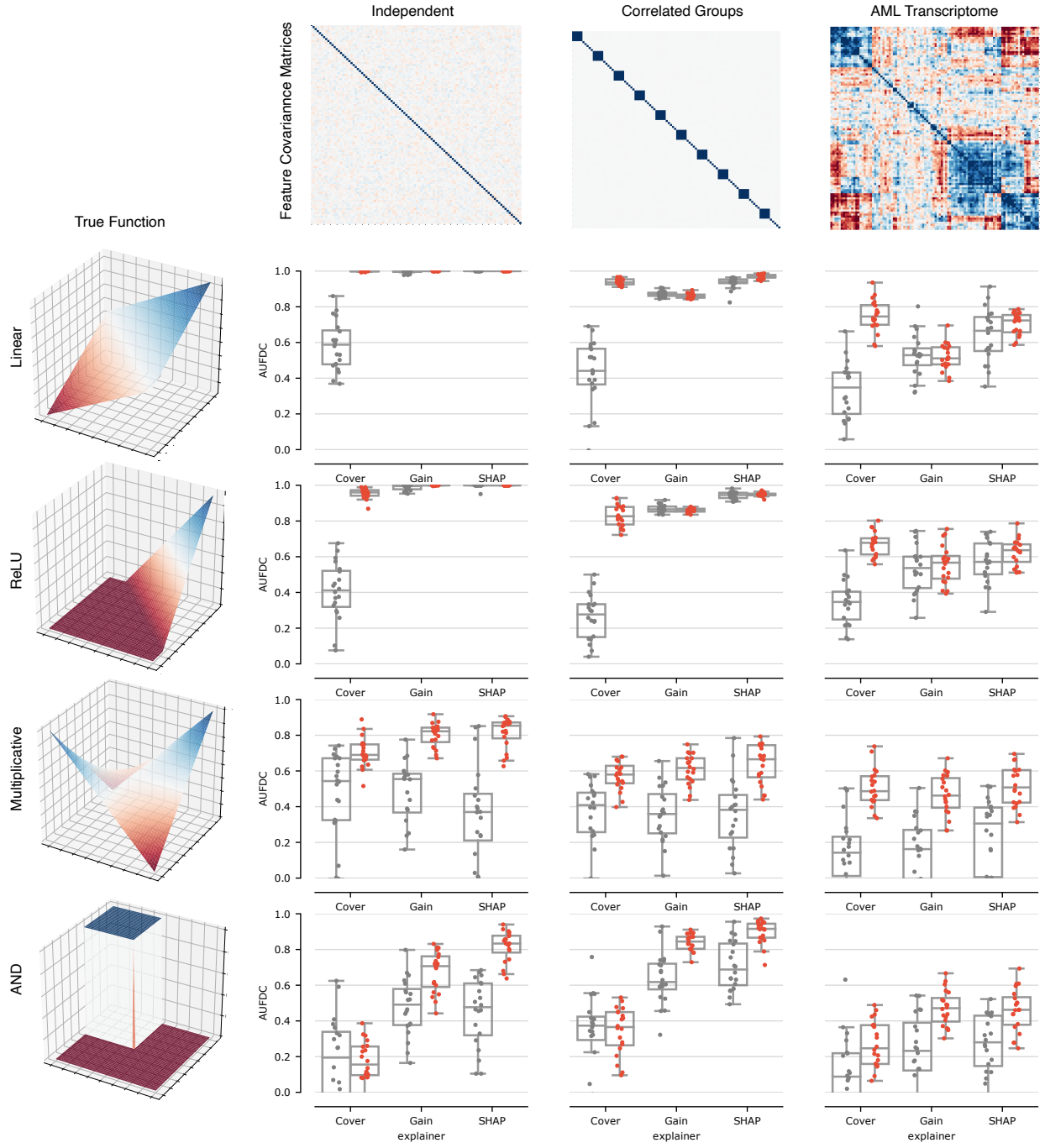

Figure 2.12: **EXPRESS improves XGBoost attributions for a variety of attribution methods.** Comparison of feature discovery performance between individual XGBoost models and ensembles of XGBoost models across all 12 dataset types from the synthetic benchmark. Three separate feature attribution methods are tested for each model: a) Cover, b) Gain, and c) SHAP.

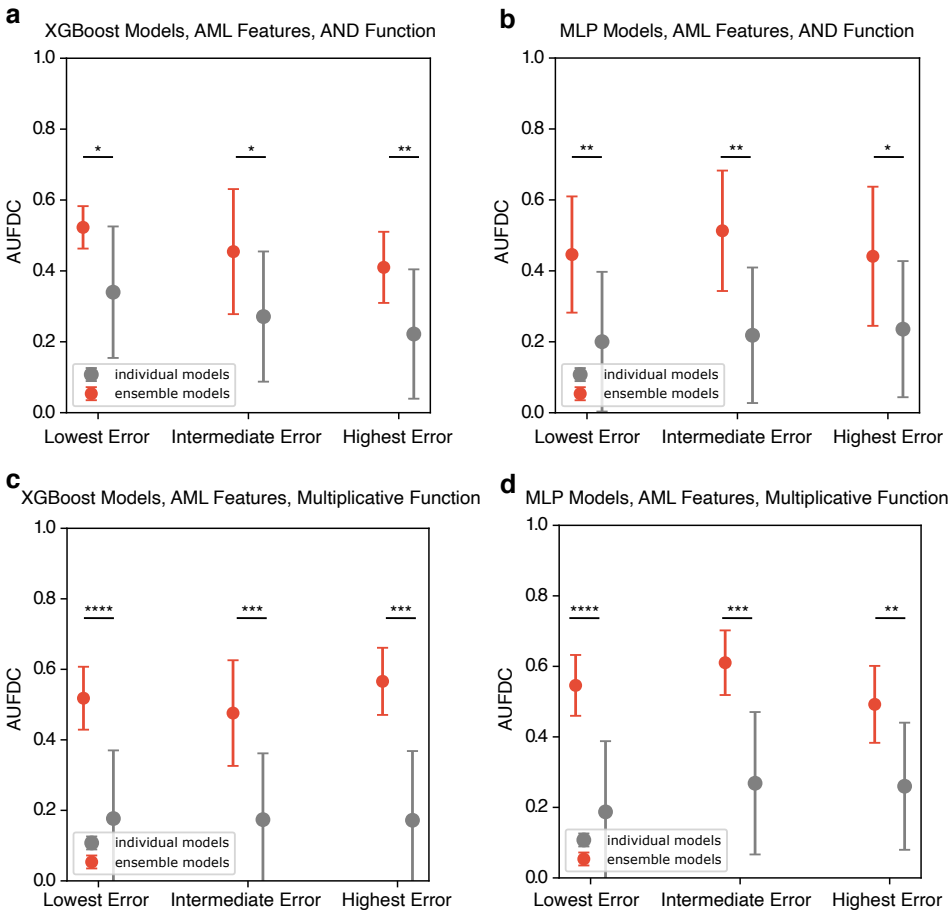

Figure 2.13: **EXPRESS improves feature attributions independently of improvement in model performance.** For both XGBoost models (**a**,**c**) and deep learning models (**b**,**d**), we see that even after controlling for the effect of model ensembles on predictive performance by stratifying models (low, intermediate, and high predictive performance), within each stratification ensemble models have significantly higher AUFDC. Significance asssed by Mann-Whitney U-test, * represents $p < 0.05$, ** represents $p < 0.01$, *** represents $p < 0.001$, and **** represents $p < 0.0001$. The circle for each plot represents the mean AUFDC of individual or ensemble models within a given predictive performance stratification, while the bars indicate plus or minus one standard deviation.

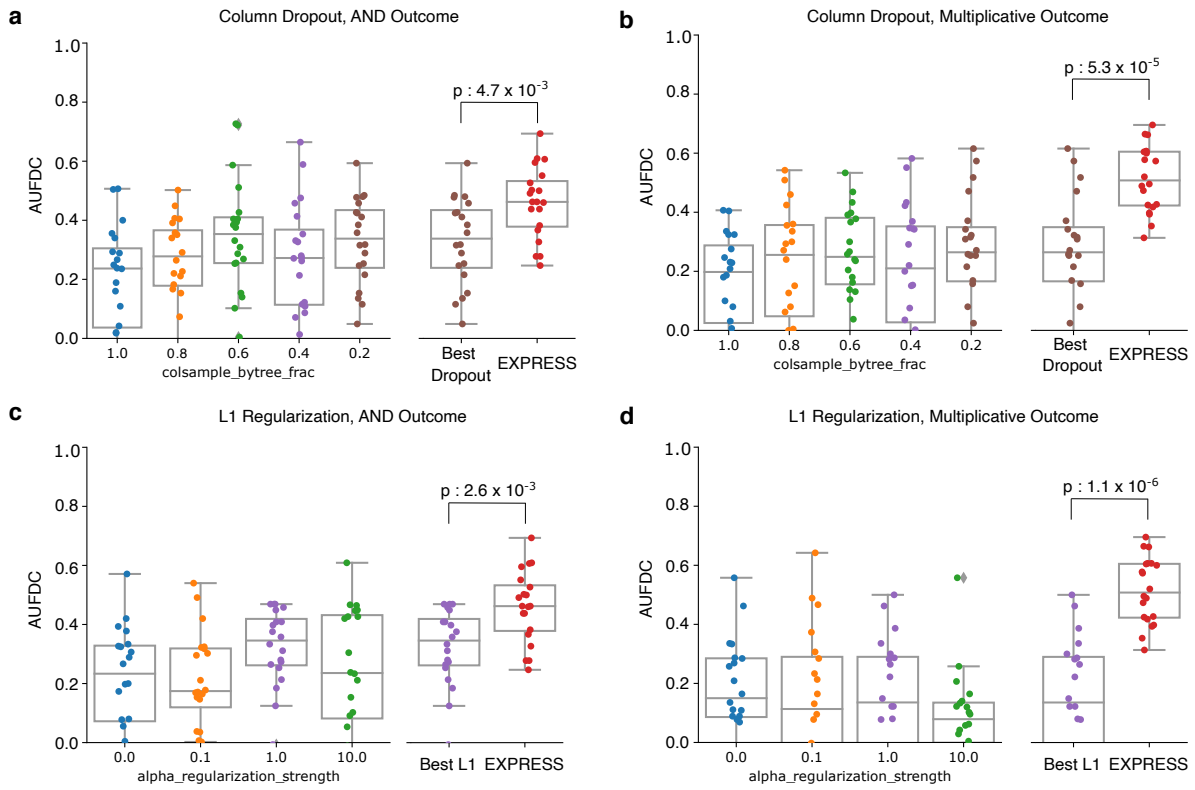

Figure 2.14: **Ensembling improves XGBoost attributions more than explicit regularization.** Using the synthetic datasets with real AML gene expression features, we compare the increase in AUFDC seen with explicit regularization, such as per-tree column dropout and L1 regularization, with ensembling. For the synthetic datasets with AML features and the AND true function, we see that ensembles improve AUFDC significantly more than column dropout (**a**, Mann-Whitney U-test, $U = 2.83$, $p = 4.7 \times 10^{-3}$) and L1 regularization (**c**, $U = 3.00$, $p = 2.7 \times 10^{-3}$). For the synthetic datasets with AML features and the multiplicative true function, we see that ensembles improve AUFDC significantly more than column dropout (**b**, $U = 4.04$, $p = 5.25 \times 10^{-5}$) and L1 regularization (**d**, $U = 4.87$, $p = 1.12 \times 10^{-6}$).

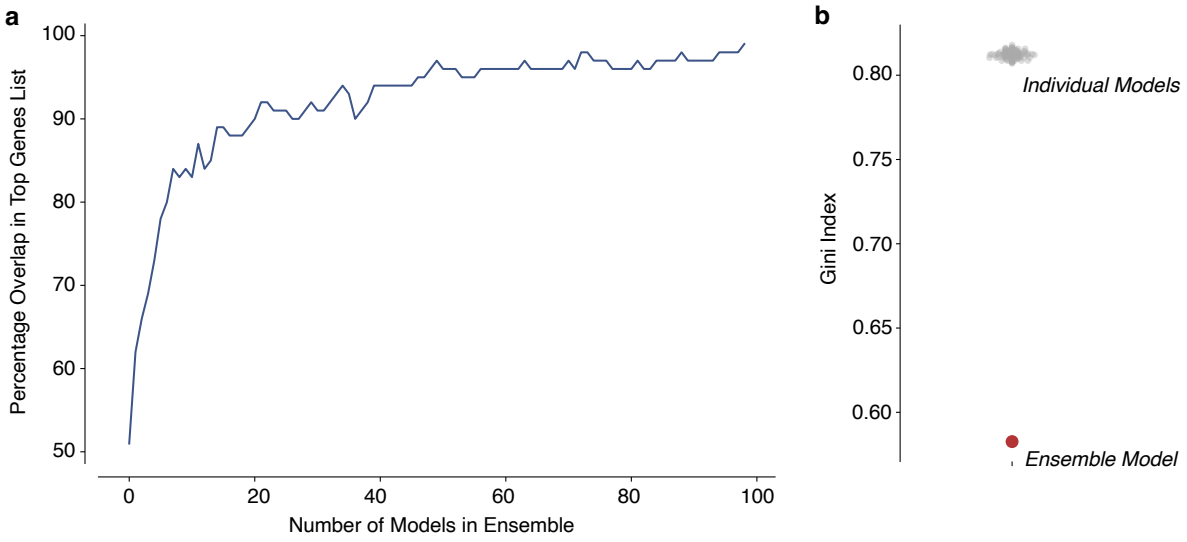

Figure 2.15: **Beat AML data attribution characteristics. a**, To ensure that a sufficient number of models were included in our final ensemble, we measured the percentage overlap in the final list of top 100 genes, and the cumulative top 100 genes list as additional XGBoost models were ensembled. **b** Attribution vector sparseness for individual models trained on Beat AML dataset (grey) and attribution vector sparseness for our final ensemble model (red). A lower Gini Index indicates a more sparse attribution vector.

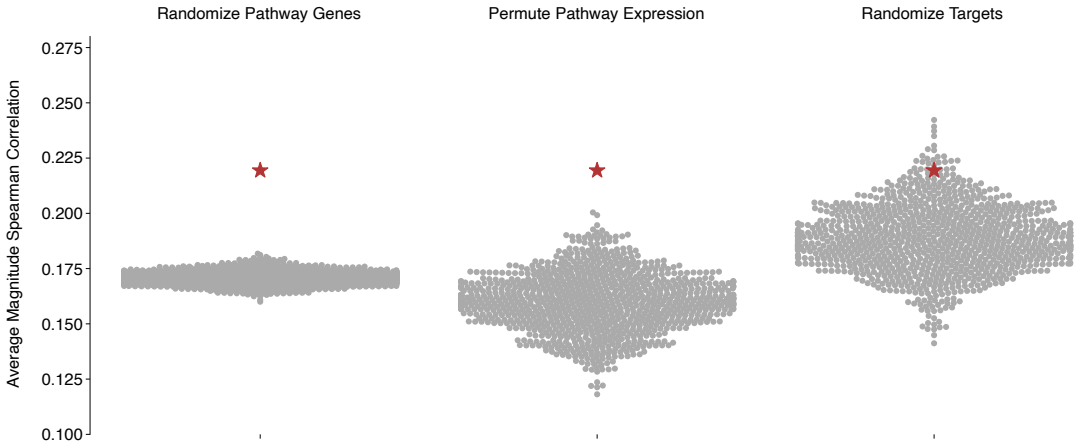

Figure 2.16: **Hematopoietic differentiation expression signature replicates in external dataeset.** Using AML cancer cell line expression data and CRISPR genetic dependency scores from the DepMap database, we show that the average magnitude association of the S1 and D1 expression signatures with genetic dependency of the targets of the drugs in Fig. 2.5f (shown as red stars) are stronger than the average magnitude associations under three separate null distributions. The first null distribution ("Randomize Pathway Genes") averages random genes rather than the actual genes in signatures S1 and D1. The second null distribution ("Permute Pathway Expression") permutes the rows in the expression matrix. The third null distribution ("Randomize Targets") measures the associations with random genetic dependency targets, rather than the actual targets of the drug combinations in Fig. 2.5f. Across all three null models, we find that the true expression signature is significantly associated with cancer cell line genetic dependency on the drug targets of the drugs in Fig. 2.5f (empirical p-values 0.001, 0.001, and 0.026, respectively).

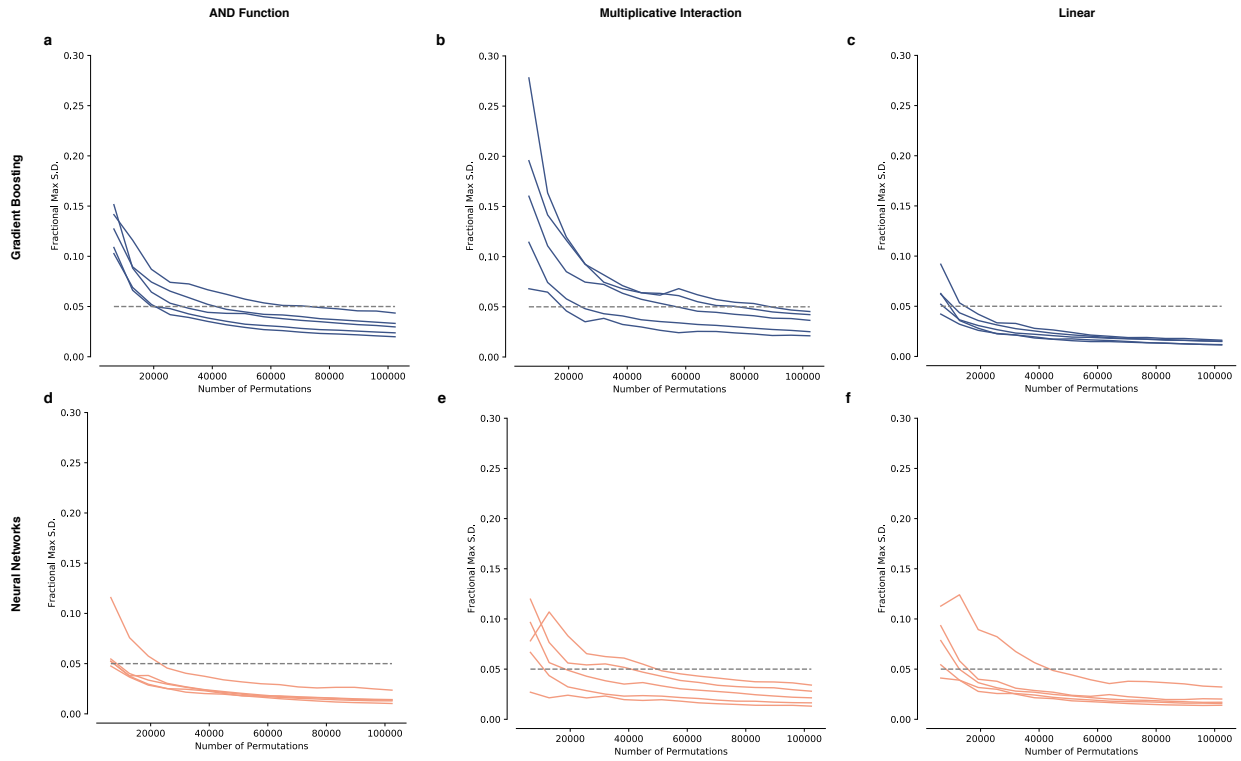

Figure 2.17: **Sampling convergence of SAGE values.** To ensure convergence of the Shapley value estimates used in our benchmark, we measured the maximum standard deviation of the elements in our attribution vector (generated using SAGE) as a fraction of the total attribution (Fractional Max S.D.), plotted against the number of permutations sampled. We found that for all three outcome types (a step function generated using a boolean AND function, a sum of pairwise multiplicative interactions, and a linear function), for both gradient boosting models (**a-c**) and neural networks (**d-f**), 102400 permutations were sufficient to drive the Fractional Max S.D. below 5%.

Chapter 3

EXPLAINING EXPLANATIONS: AXIOMATIC FEATURE INTERACTIONS FOR DEEP NETWORKS

3.1 Introduction and Prior Work

Deep neural networks are one of the most popular classes of machine learning (ML) model. They can achieve state-of-the-art performance in problem domains ranging from natural language processing to image recognition [56, 99]. They have even outperformed other non-linear model types on structured tabular data [253]. Because neural networks have traditionally been more difficult to interpret than simpler model classes, gaining a better understanding of their predictions is desirable for many reasons. Where these algorithms are used in automated decisions that affect humans, explanations may be legally required [249]. When used in high stakes applications, it is essential to ensure that models are making safe decisions for the right reasons [80]. During model development, interpretability methods can help debug undesirable model behavior [274].

3.1.1 Feature Attribution Methods

Many recent approaches focus on interpreting deep neural networks, ranging from those aiming to distill complex models into simpler ones [279, 300, 222] to those seeking to identify key concepts learned by a network [125, 207, 206, 72, 64, 185]. Other approaches aim to build interpretable deep models, such as probabilistic interpretable deep models and deep networks for causal effects [306, 312, 173]. Among the best-studied sets of approaches for interpreting deep neural networks is a class known as *feature attribution methods* [21, 256, 179, 230]. These approaches explain a model's prediction by assigning credit to each input feature based on how much it influences the prediction. Although these approaches help practitioners identify salient features, they do not explain *why* the features are important or address features interactions in a model. To enrich our understanding of model behavior, we must develop methods to explain both. For example, in Figure 3.1, we show that word-level interactions can help us distinguish why deeper, more expressive neural networks outperform simpler ones on language tasks.

3.1.2 Feature Interaction Methods

Several existing methods explain feature interactions in neural networks. [48] explain global interactions in Bayesian Neural Networks (BNN) by examining pairs of features that have large second-order derivatives at the input. The Neural Interaction Detection method detects statistical interactions between features by examining the weight matrices of feed-forward neural networks [283]. Further, several authors have proposed domain-specific methods to find interactions in the area of deep learning for genomics [136, 89]. For example, Deep Feature Interaction Maps detect interactions between two features by calculating the change in the attribution of one feature that is incurred by changing the value of the second [89]. [260] generalize Contextual Decomposition [195] to explain interactions for feed-forward and convolutional architectures. In game theory literature, [87] propose the Shapley Interaction Index, which allocates credit to interactions between players in a coalitional game by considering all possible subsets of players. Recently, [57] suggested a modified version of the Shapley Interaction Index that weights certain subsets of players differently in order to achieve a different set of desired axioms.

A recent paper by [284] proposes a method for "interaction attribution," which they compare to our method. Their approach has three steps: (1) detect pairwise interactions between features using a method called ArchDetect, (2) use these pairwise interactions to cluster features into groups so that interactions occur only between features within the same group, and (3) attribute importance to those groups of features using a method called ArchAttribute. In Section 3.9.1, we show that the quantity described by ArchAttribute is equivalent to a *Group Shapley value* rather than an interaction value, a fundamentally different quantity than we aim to measure in our paper, and hence is not comparable. Our method detects the extent of non-additivity between feature pairs, while their approach quantifies the total contribution of a group of interacting features. Both quantities may be of interest, depending on the particular application. Therefore, we instead compare our interaction detection method to ArchDetect.

3.1.3 Limitations of Prior Approaches

Previous approaches have substantially advanced our understanding of feature interaction in neural networks. However, all suffer from practical limitations, including being limited to specific types of architectures. Neural Interaction Detection applies only to feed-forward neural network architectures and cannot be used on networks with convolutions, recurrent units, or self-attention. Contextual Decomposition has been applied to LSTMs, feed-forward neural networks and convolutional networks, but to our knowledge is not straightforward to apply to more recent innovations in deep learning, such as self-attention layers. The approach suggested by [48] is limited by its required use of Bayesian Neural Networks; it is unclear

how to apply the method to standard neural networks. Deep Feature Interaction Maps work when a model's input features have only a few discrete values (such as genomic sequence data, which can be *in silico* mutated), since it marginalizes over all possible values of paired input features. The Shapley Interaction Index and Shapley Taylor Interaction Index, like the Shapley value, are NP-hard to compute exactly [63].

Furthermore, most current methods to detect interactions fail to satisfy the common-sense axioms proposed for feature attribution methods [274, 179]. As a result, these approaches are provably unable to find learned interactions or more generally find counter-intuitive interactions (see Section 3.4). Current methods that do satisfy such axioms, such as those based on the Shapley Interaction Index [87, 57], are computationally inefficient to compute or even approximate.

3.1.4 Our contributions

First, we propose an approach, Integrated Hessians, to quantify pairwise feature interactions that can be applied to any neural network architecture. We also identify several common-sense axioms that feature-level interactions should satisfy and show that our proposed method satisfies them. Third, we provide a principled way to compute interactions in ReLU-based networks, which are piece-wise linear and have zero second derivatives. Further, we evaluate our method against existing methods and show that it more accurately identifies interactions in simulated data. Finally, we demonstrate the utility of Integrated Hessians in a variety of applications where identifying feature interactions in neural networks is useful.

3.2 Explaining Explanations with Integrated Hessians

To derive our feature interaction values, we first consider Integrated Gradients (IG), a feature attribution method proposed by [274] based on the Aumann-Shapley value; the Aumann-Shapley value is a variant of the Shapley value for cooperative games with continuous rather than discrete players [8]. We represent our model as a function $f : \mathbb{R}^d \mapsto \mathbb{R}$.[1] For a function $f(x)$, the IG attribution for the ith feature is defined as:

$$\phi_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha, \tag{3.1}$$

where x is the sample to be explained and x' is a baseline value. For notation concision, we suppress the dependence on the choice of baseline x' when we write the IG attribution $\phi_i(x)$, but we discuss the importance of baselines in Section 3.2.1 and Section 3.9.4. Although

[1] For multi-output models, such as multi-class classification problems, we assume the function is indexed into the correct output class.

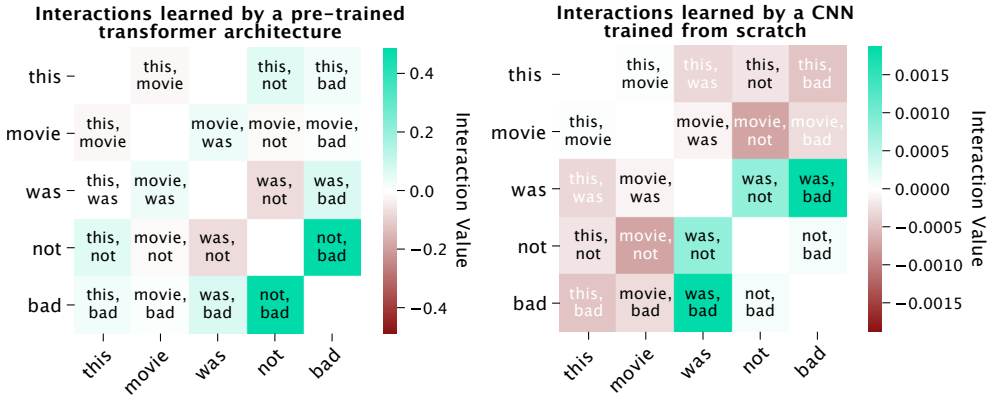

Figure 3.1: Interactions explain why certain models outperform others. Here, we examine word interactions in the sentence "this movie was not bad." We compare two models trained to perform sentiment analysis on the Stanford Sentiment data set: a pre-trained transformer, DistilBERT (left), which predicts the sentence has a positive sentiment with 98.2% confidence, and a convolutional neural network trained from scratch (right), which predicts a negative sentiment with 97.6% confidence. The transformer picks up negation patterns: "not bad" has a positive interaction, despite the word "bad" being negative. The CNN mostly picks up negative interactions, like "movie not" and "movie bad."

f is often a neural network, the sole requirement for computing attribution values is that f be differentiable along the path from x' to x. Our key insight is that the IG value for a differentiable model $f : \mathbb{R}^d \mapsto \mathbb{R}$ is *itself* a differentiable function $\phi_i : \mathbb{R}^d \mapsto \mathbb{R}$. This means that we can apply IG to itself in order to explain the degree to which feature j impacted the importance of feature i:

$$\Gamma_{i,j}(x) = \phi_j(\phi_i(x)). \tag{3.2}$$

For $i \neq j$, we derive that:

$$\Gamma_{i,j}(x) = (x_i - x'_i)(x_j - x'_j) \times \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \alpha\beta \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\alpha d\beta. \tag{3.3}$$

For $i = j$, the formula $\Gamma_{i,i}(x)$ has an additional first-order term:

$$\begin{aligned} \Gamma_{i,i}(x) = (x_i - x'_i) \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha\beta(x - x'))}{\partial x_i} d\alpha d\beta + \\ (x_i - x'_i)^2 \times \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \alpha\beta \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\alpha d\beta. \end{aligned} \tag{3.4}$$

We interpret $\Gamma_{i,j}(x)$ as the explanation of the importance of feature i in terms of the input value of feature j. For a full derivation of Integrated Hessians, see Section 3.8.

3.2.1 Baselines and Expected Hessians

Several feature attribution methods have identified the necessity of a baseline value representing a lack of information when generating explanations [274, 256, 21, 179]. However, more recent work notes that choosing a single baseline value to represent a lack of information can be challenging in certain domains [129, 120, 273, 5, 73, 268]. As an alternative, [65] proposed an extension of IG called Expected Gradients (EG), which samples many baseline inputs from the training set. We can therefore apply EG to itself to get Expected Hessians:

$$\Gamma^{EG}_{i,j}(x) = \mathbb{E}_{\alpha\beta\sim U(0,1)\times U(0,1),x'\sim D}\left[(x_i - x'_i)(x_j - x'_j)\alpha\beta\frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j}\right]. \tag{3.5}$$

$$\begin{aligned}\Gamma^{EG}_{i,i}(x) = \mathbb{E}_{\alpha\beta\sim U(0,1)\times U(0,1),x'\sim D}\Bigg[&(x_i - x'_i)\frac{\partial f(x' + \alpha\beta(x - x'))}{\partial x_i} \\ &+ (x_i - x'_i)^2\alpha\beta\frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j}\Bigg],\end{aligned} \tag{3.6}$$

where the expectation is over $x' \sim D$ for an underlying data distribution D, $\alpha \sim U(0,1)$ and $\beta \sim U(0,1)$. This formulation is useful when there is no single, natural baseline. Deriving Expected Hessians follows the same steps as deriving Integrated Hessians, but the integrals can be viewed as integrating over the product of two uniform distributions $\alpha\beta \sim U(0,1) \times U(0,1)$. We use Integrated Hessians for all main text examples; however, some of the supplementary examples use the Expected Hessians formulation.

Almost every attribution and interaction method to date requires a choice of baseline. Some approaches explicitly discuss their reliance on baselines [274, 272, 284], while others rely on baselines without explicitly mentioning them in their work. For example, Contextual Decomposition turns features "off" by setting them to a single reference baseline of 0 [195]. By implementing a solution for domains where it is difficult to choose a single baseline, we seek to alleviate issues common to many interaction methods. Additionally, we note that the formulation as an expectation strictly generalizes the Integrated Hessians formulation; while it is possible to sample from many different baselines, integrating from a single one can also be conceptualized as sampling from a distribution with all of its density on that baseline. We do not always anticipate that sampling a larger distribution of baselines would be advantageous, especially in application settings where a single natural baseline can be easily defined.

3.2.2 Fundamental Axioms for Interaction Values

We now describe common-sense axioms that every interaction method should satisfy, and we show that Integrated Hessians satisfies them all.

Self and Interaction Completeness Axiom

[274] showed that, among other theoretical properties, IG satisfies the completeness axiom, which states: $\sum_i \phi_i(x) = f(x) - f(x')$. We show the following two equalities, which are immediate consequences of completeness:

$$\sum_i \sum_j \Gamma_{i,j}(x) = f(x) - f(x'). \tag{3.7}$$

$$\Gamma_{i,i}(x) = \phi_i(x) - \sum_{j \neq i} \Gamma_{i,j}(x). \tag{3.8}$$

We call equation (3.7) the *interaction completeness* axiom: the sum of the $\Gamma_{i,j}(x)$ terms adds up to the difference between the output of f at x and at the baseline x'. This axiom lends itself to another natural interpretation of $\Gamma_{i,j}(x)$: as the interaction between features i and j. That is, it represents the contribution that the pair of features i and j together add to the output $f(x) - f(x')$. It is vital to satisfy interaction completeness because it demonstrates a relationship between model output and interaction values. Without this axiom, it would not be clear how to interpret the scale of interactions.

Equation (3.8) shows how to interpret the self-interaction term $\Gamma_{i,i}(x)$ as the *main effect* of feature i after interactions with all other features have been subtracted. We note that equation (3.8) also implies the following, intuitive property about the main effect: if $\Gamma_{i,j} = 0$ for all $j \neq i$, or in the degenerate case where i is the only feature, we have $\Gamma_{i,i} = \phi_i(x)$. We call this the *self-completeness* axiom. Satisfying self-completeness provides the vital guarantee that the main effect of feature i equals its feature attribution value if that feature interacts with no other features.

The proof of these two equations is straightforward. First, we note that $\phi_i(x') = 0$ for any i because $x'_i - x'_i = 0$. Then, by completeness of Integrated Gradients, we have that:

$$\sum_j \Gamma_{i,j}(x) = \phi_i(x) - \phi_i(x') = \phi_i(x). \tag{3.9}$$

Re-arrangement provides the *self-completeness* axioms:

$$\Gamma_{i,i}(x) = \phi_i(x) \text{ if } \Gamma_{i,j}(x) = 0, \forall j \neq i. \tag{3.10}$$

Since Integrated Gradients satisfies completeness:

$$\sum_i \phi_i(x) = f(x) - f(x'). \tag{3.11}$$

Making the appropriate substitution from equation 3.9 shows the *interaction completeness* axiom:

$$\sum_i \sum_j \Gamma_{i,j}(x) = f(x) - f(x'). \tag{3.12}$$

Sensitivity Axiom

Integrated Gradients satisfies an axiom called *sensitivity*, which states that given an input x and a baseline x', if $x_i = x'_i$ for all i except j where $x_j \neq x'_j$ and if $f(x) \neq f(x')$, then $\phi_j(x) \neq 0$. Specifically, by completeness we know that $\phi_j(x) = f(x) - f(x')$. Intuitively, this means that if only one feature differs between the baseline and the input and changing that feature would change the output, then the amount the output changes should be equal to the importance of that feature.

We can extend this axiom to interactions by considering the case where two features differ from the baseline. We call this axiom *interaction sensitivity* and describe it as follows. If an input x and a baseline x' are equal everywhere except $x_i \neq x'_i$ and $x_j \neq x'_j$, and if $f(x) \neq f(x')$, then: $\Gamma_{i,i}(x) + \Gamma_{j,j}(x) + 2\Gamma_{i,j}(x) = f(x) - f(x') \neq 0$ and $\Gamma_{\ell,k} = 0$ for all $\ell, k \neq i, j$. Intuitively, this states that if the only features that differ from the baseline are i and j, then the difference in the output $f(x) - f(x')$ must be solely attributable to the main effects of i and j plus the interaction between them. This axiom holds simply by applying *interaction completeness* and observing that $\Gamma_{\ell,k}(x) = 0$ if $x_\ell = x'_\ell$ or $x_k = x'_k$.

Implementation Invariance Axiom

The implementation invariance axiom, described in [274], states that for two models f and g such that $f = g$, then $\phi_i(x; f) = \phi_i(x; g)$ for all features i and all points x regardless of how f and g are implemented. Although seemingly trivial, this axiom does not necessarily hold for attribution methods that use the network's implementation or structure to generate attributions. Critically, this axiom also does not hold for the interaction method proposed by [283], which looks at the first layer of a feed-forward neural network. Two networks may represent exactly the same function but differ greatly in their first layer.

This axiom is trivially seen to hold for Integrated Hessians since it holds for Integrated Gradients. However, without this key axiom, attributions/interactions could encode information about unimportant aspects of model structure rather than the actual decision surface of the model.

Linearity Axiom

Integrated Gradients also satisfies an axiom called *linearity*. Given two networks f and g, consider the output of the weighted ensemble of both networks to be $af(x)+bg(x)$. Then, the attribution $\phi_i(x; af + bg)$ of the weighted ensemble equals the weighted sum of attributions $a\phi_i(x; f) + b\phi_i(x; g)$ for all features i and samples x. This important axiom preserves network linearity and facilitates easy computation of attributions for network ensembles.

We can generalize linearity to interactions using the *interaction linearity* axiom:

$$\Gamma_{i,j}(x; af + bg) = a\Gamma_{i,j}(x; f) + b\Gamma_{i,j}(x; g), \tag{3.13}$$

for any i, j and all points x. Given that $\Gamma_{i,j}$ is a composition of linear functions ϕ_i, ϕ_j in terms of the parameterized networks f and g, it is itself a linear function of the networks; therefore, Integrated Hessians satisfies *interaction linearity*.

Symmetry-Preserving Axiom

We say that two features x_i and x_j are *symmetric* with respect to f if swapping them does not change the output of f anywhere. That is, $f(\cdots, x_i, \cdots, x_j, \cdots) = f(\cdots, x_j, \cdots, x_i, \cdots)$. [274] shows that Integrated Gradients is *symmetry preserving*, that is, if x_i and x_j are symmetric with respect to f, and if $x_i = x_j$ and $x'_i = x'_j$ for some input x and baseline x', then $\phi_i(x) = \phi_j(x)$. To generalize to interaction values, if the same conditions as above hold, then $\Gamma_{k,i}(x) = \Gamma_{k,j}(x)$ for any feature x_k. This axiom, which holds since $\Gamma_{k,i}(x) = \phi_i(\phi_k(x))$ and ϕ_i, ϕ_j are symmetry-preserving, is significant because it indicates that if two features are functionally equivalent to a model, then they must interact in the same way with respect to that model.

Interaction Symmetry Axiom

Another form of symmetry is important to note: *interaction symmetry*. This axiom states that, for any i, j, we have $\Gamma_{i,j}(x) = \Gamma_{j,i}(x)$. Although simple, it guarantees that the interaction function itself is symmetric with respect to the features it explains. It is straightforward to show that existing neural networks and their activation functions have continuous second partial derivatives, which implies that Integrated Hessians satisfies interaction symmetry.[2]

3.2.3 Approximating Integrated Hessians in Practice

Our interaction values include a double integral, which is intractable to compute analytically in the general case. To compute Integrated Gradients in practice, [274] introduced the

[2]Section 3.3 describes the special case of the ReLU activation function.

following discrete sum approximation:

$$\hat{\phi}_i(x) = (x_i - x'_i) \times \sum_{\ell=1}^{k} \frac{\partial f(x' + \frac{\ell}{k}(x - x'))}{\partial x_i} \times \frac{1}{k}, \tag{3.14}$$

where k is the number of points used to approximate the integral. To compute Integrated Hessians, we introduce a similar discrete sum approximation:

$$\hat{\Gamma}_{i,j}(x) = (x_i - x'_i)(x_j - x'_j) \times \sum_{\ell=l}^{k} \sum_{p=1}^{m} \frac{\ell}{k} \times \frac{p}{m} \times \frac{\partial f(x' + (\frac{\ell}{k} \times \frac{p}{m})(x - x'))}{\partial x_i \partial x_j} \times \frac{1}{km}. \tag{3.15}$$

Typically, it is easiest to compute this quantity when $k = m$ and the number of samples drawn is thus a perfect square. However, when a non-square number of samples is preferred we can generate sample points from the product distribution of two uniform distributions so that the number is the largest perfect square above the desired number of samples; we can index the sorted samples appropriately to get the desired number. The preceding formula omits the first-order term in $\Gamma_{i,i}(x)$, but it can be computed using the same principle.

Expected Hessians has a similar, if slightly easier, form:

$$\hat{\Gamma}^{EG}_{i,j}(x) = (x_i - x'_i)(x_j - x'_j) \sum_{\ell}^{k} \zeta_\ell \times \frac{\partial f(x' + \zeta_\ell(x - x'))}{\partial x_i \partial x_j} \times \frac{1}{k}, \tag{3.16}$$

where ζ_ℓ is the ℓth sample from the product distribution of two uniform distributions. We find in general that less than 300 samples are required for any given problem to approximately satisfy interaction completeness. For most problems, far less than 300 suffices (e.g., around 50), but the number is model and data dependent: larger models and higher-dimensional data generally require more samples than smaller models and lower-dimensional data.

3.3 Smoothing ReLU Networks

One major limitation not discussed in previous approaches to interaction detection in neural networks relates to use of the ReLU activation function, $\text{ReLU}(x) = \max(0, x)$, in many popular neural network architectures. Neural networks using ReLU are piecewise linear and have second partial derivatives equal to zero in all places. Previous second-order approaches (based on the Hessian) fail to detect any interaction in ReLU-based networks.

Fortunately, the ReLU activation function has a smooth approximation – the SoftPlus function: $\text{SoftPlus}_\beta(x) = \frac{1}{\beta}\log(1 + e^{-\beta x})$. SoftPlus more closely approximates ReLU as β

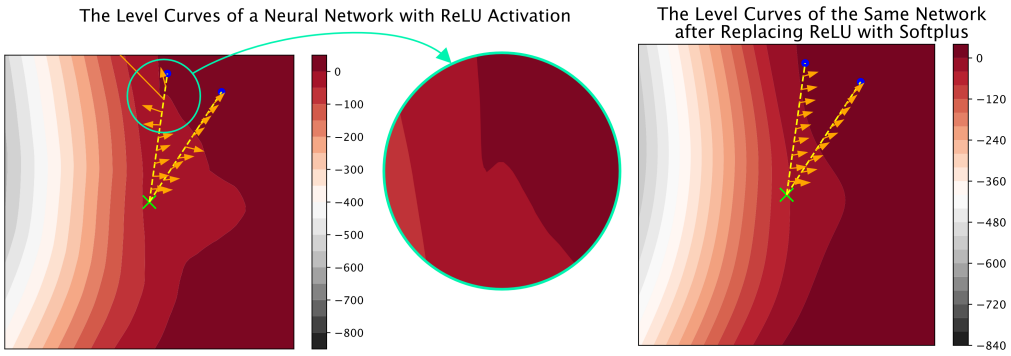

Figure 3.2: Replacing ReLU activations (left) with SoftPlus$_\beta$ activations (right) where $\beta = 10$ smooths the decision surface of a neural network: gradients tend to be more homogeneous along the integration path. Orange arrows show the gradient vectors at each point along the path from the reference (green x) to the input (blue dots). ReLUs can cause small bends in the output space with aberrant gradients.

increases and has well-defined higher-order derivatives. Furthermore, [59] have proved that a model's outputs *and* first-order feature attributions are minimally perturbed when ReLU activations are replaced by SoftPlus activations in a trained network. Therefore, we can apply Integrated Hessians on a network with ReLU activations by first replacing ReLU with SoftPlus. We note that no re-training is necessary for this approach.

In addition to being twice differentiable and letting us calculate interaction values in ReLU networks, replacing ReLU with SoftPlus offers other benefits when calculating interaction values. We show that smoothing a neural network (i.e., decreasing the value of β in the SoftPlus activation function) lets us accurately approximate the Integrated Hessians value with fewer gradient calls.

Theorem 1. *For a one-layer neural network with softplus$_\beta$ non-linearity, $f_\beta(x) = \text{softplus}_\beta(w^T x)$, and d input features, we can bound the number of interpolation points k needed to approximate the Integrated Hessians to a given error tolerance ϵ by $k \leq \mathcal{O}(\frac{d\beta^2}{\epsilon})$.*

The proof of 1 is shown in Section 3.10. In addition to the proof for the single-layer case, Fig. 3.10-3.11 also presents empirical results to show that many-layered neural networks display the same property.

The intuition behind these results is that as we replace ReLU with SoftPlus, the decision surface of the network is smoothed (see Fig. 3.2). We observe that the gradients tend to all have more similar direction along the path from reference to foreground sample once the network has been smoothed with SoftPlus replacement.

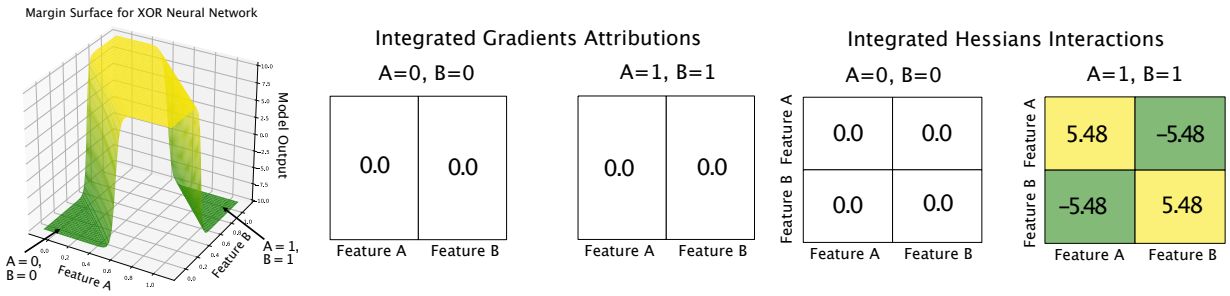

Figure 3.3: (Left) The margin surface for a neural network representing an XOR function. (Middle) Integrated Gradients feature attributions for two samples, the first where both features are turned off, and the second where both features are turned on. Both get identical Integrated Gradients attributions. (Right) Integrated Hessians feature interactions for the same two samples. We see that the Integrated Hessians values, but not the Integrated Gradients values, differentiate the two samples.

3.4 Explanation of XOR function

To explain why feature interactions can be more informative than feature attributions, we use the case of two binary features and a neural network representing an XOR function. This network has a large output when either feature is on alone, but it has a low magnitude output when both features are either on or off (Fig. 3.3, left).

When we explain the network using Integrated Gradients with the zeros baseline, we see that samples where both features are on and those where both features are off get identical attributions (see Fig. 3.3, middle). Integrated Hessians, however, differentiates these two samples by identifying the negative interaction that occurs between the two features when they are both on (Fig. 3.3, right). Therefore, the interactions can usefully distinguish between (0,0), which has an output of 0 because it is identical to the baseline, and (1,1), which has an output of 0 because both features are on. When considered independently, each feature would increase the model's output, but when considered in interaction with one another, they cancel out the positive effects and drive the model's output back to the baseline.

This example also illustrates a problem with methods like [48], which use the input Hessian without integrating over a path. In Fig. 3.3, we see that the function is saturated at all points on the data manifold, meaning all elements of the Hessian will be 0 for all samples. In contrast, by integrating between the baseline and the samples, Integrated Hessians can correctly detect the negative interaction between the two features.

3.5 Empirical Evaluation

We empirically evaluated our method against other methods using benchmarks inspired by recent literature on quantitatively evaluating feature attribution methods [3, 129, 101, 302, 163]. We compare Integrated Hessians to six existing methods: the Shapley Interaction Index [87], the Shapley Taylor Interaction Index [57], ArchDetect [284], Generalized Contextual Decomposition [260], Neural Interaction Detection [283], and using the Hessian at the input sample [48].

3.5.1 Computation Time

First, we compared the computation time of each method. We explained interactions in a 5-layer neural network with SoftPlus activation and 128 units per layer. We ran each method on models with 5, 50 and 500 input features, and evaluated each method on 1000 samples. Because computing the Shapley Interaction Index and Shapley Taylor Interaction Index is NP-hard in the general case [63], we instead compared against a Monte Carlo estimation of the Shapley Interaction Index with 200 samples, analogous to how the Shapley Value is estimated in [135]. Even using a small number of samples, this comparison could not run to completion for the 500 feature case and would have taken an estimated *1000 hours* to complete. For each method, we computed all pairwise interactions: d^2 interactions for d features. Figure 3.4 (left) shows results.

We observe that our method is more tractable than the two other axiomatic approaches to interaction based on Shapley values. Further, we note that as the number of features grows, our method is more tractable than several of the heuristic methods, as well: other methods require at least $O(d^2)$ separate forward passes of the model to compute the interactions, which is non-trivial to parallelize. However, back-propagating the Hessian through the model is easily done in parallel on a GPU since this functionality already exists in modern deep learning frameworks [211, 2].

3.5.2 Quantitative Comparison of Interaction Detection with Other Methods

To compare each method, we used the Remove and Retrain benchmark introduced in [101]. The benchmark compares feature attributions by progressively ablating the most important features in each sample —ranked according to each attribution method —and then retraining the model on the ablated data and measuring the performance drop. The attribution method that incurred the fastest performance decline was the quickest to identifythe most predictive features in the data.

We generated a simulated regression task with 10 features where each feature was drawn independently from $\mathcal{N}(0, 1)$. The label was an additive sum of 20 interactions with random

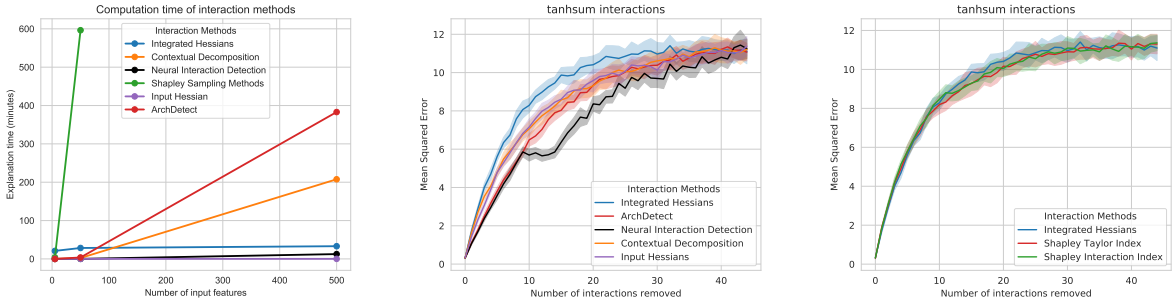

Figure 3.4: *Left*: The time each method takes to compute all pairwise interactions on 1000 samples with d features as a function of d. Existing methods scale poorly compared to our proposed method. We also benchmarked our methods against others (*Center*, heuristic methods; *Right*, Shapley value-based methods) using a modified version of Remove and Retrain [101] on simulated interactions. Our method more accurately identifies the most important interactions than all existing methods other than the Shapley Interaction Index and Shapley Taylor Index, which are much more computationally expensive.

coefficients normalized to sum to 1, drawn without replacement from all possible pairs of features. A 3-layer neural network's predictions achieved over 0.99 correlation with the true label.

We compared each of the five interaction methods using Remove and Retrain on five different interaction types: $g_{\text{tanhsum}}(x_i, x_j) = \tanh(x_i + x_j)$, $g_{\text{cossum}}(x_i, x_j) = \cos(x_i + x_j)$, $g_{\text{multiply}}(x_i, x_j) = x_i * x_j$, $g_{\max}(x_i, x_j) = \max(x_i, x_j)$ and $g_{\min}(x_i, x_j) = \min(x_i, x_j)$. The results for g_{tanhsum} are displayed in Figure 3.4 (right). Our method most quickly identified the highest-magnitude interactions in the data, as demonstrated by the fastest increase in error. Our method outperformed all existing heuristic methods on all interaction types and performed equivalently to both the Shapley Interaction Index and the Shapley Taylor Interaction Index.

In addition to the Remove and Retrain benchmark, we also tested our approach using the "sanity checks" proposed in [3] to ensure that our interaction attributions were sensitive to network and data randomization. We found that our method passed both sanity checks.

3.6 Applications of Integrated Hessians

3.6.1 NLP

Over the past decade, neural networks have been the go-to model for language tasks, from convolutional [128] to recurrent [275]. More recently, large, pre-trained transformer architectures [216, 56] have achieved state-of-the-art performance on a wide variety of tasks. Previous work suggested investigating the internal weights of the attention mechanisms in attention-based models [82, 156, 164, 293]. However, more recent work suggests that examining attention weights may not be a reliable way to interpret models with attention layers [251, 110, 31]. To resolve this issue, feature attributions have been applied to text classification models to understand which words most affect classification [166, 151]. However, these methods do not explain how words interact with their surrounding context.

We downloaded pre-trained weights for DistilBERT [242] from the HuggingFace Transformers library [297]. We fine-tuned the model on the Stanford Sentiment Treebank data set [263], where the task was to predict whether a movie review had positive or negative sentiment. After 3 epochs of fine-tuning, DistilBERT achieved a validation accuracy of 0.9071 (0.9054 TPR / 0.9089 TNR).[3]

In Figure 3.5, we show interactions generated by Integrated Hessians and attributions generated by Integrated Gradients on an example drawn from the validation set. The figure demonstrates that DistilBERT learned intuitive interactions that would not be revealed from feature attributions alone. For example, a word like “painfully," which might have a negative connotation on its own, has a large positive interaction with the word “funny" in the phrase “painfully funny." In Figure 3.1, we demonstrate how interactions can help us understand why a fine-tuned DistilBERT model outperforms a simpler model: a convolutional neural network (CNN) that achieves an accuracy of 0.82 on the validation set. DistilBERT picks up positive interactions between negation words (“not”) and negative adjectives (“bad”) that a CNN fails to fully capture. Finally, in Figure 3.5, we use interaction values to reveal saturation effects: many negative adjectives describing the same noun interact positively. Although this may initially seem counter-intuitive, it reflects the structure of language. If a phrase has only one negative adjective, it stands prominently as the word that makes the phrase negative. At some point, however, describing a noun with an increasing number of negative adjectives makes any individual negative adjective less important towards classifying that phrase as negative.

[3]This performance does not represent state-of-the-art, nor is sentiment analysis representative of the full complexity of existing language tasks. However, this paper focuses on explanation, and sentiment analysis is easily fine-tuned without extensive hyperparameter search.

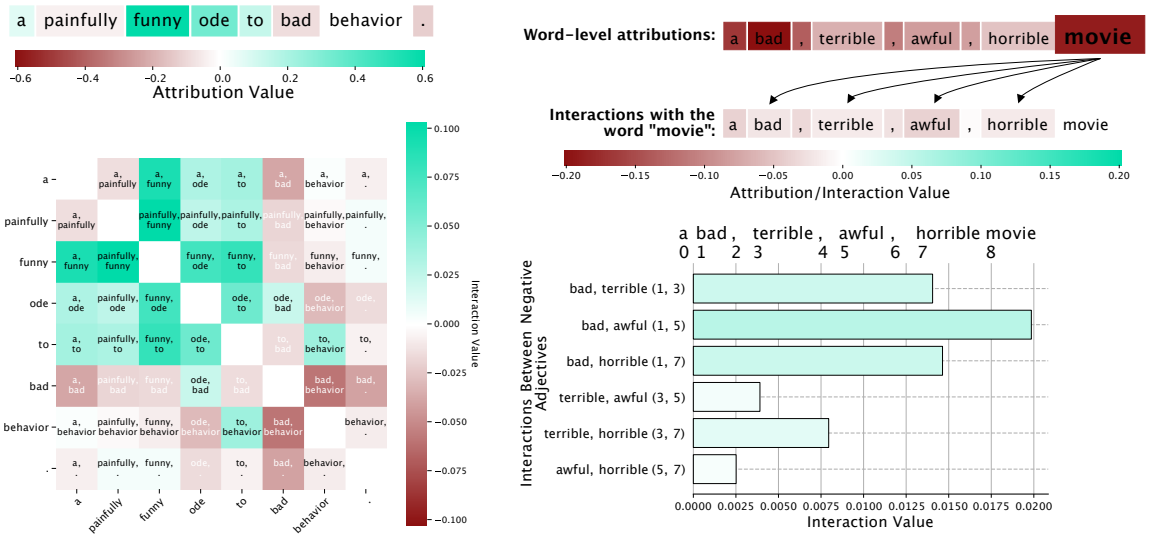

Figure 3.5: *Left*: Interactions in text reveal learned patterns such as the phrase "painfully funny" having a positive interaction despite the word "painfully" having a negative attribution. These interactions are not evident from attributions alone. *Right*: Interactions help reveal an unintuitive pattern in language models: saturation. Although the word “movie" interacts negatively with all negative modifying adjectives, those negative adjectives themselves all interact positively. The greater the number of negative adjectives in a sentence, the less each individual negative adjective contributes to the overall classification of the sentence.

3.6.2 Drug Combination Response Prediction

In the domain of anti-cancer drug combination response prediction, plotting Integrated Hessians can help to glean biological insights into the process we are modeling. We considered one of the largest publicly available data sets measuring drug combination response in acute myeloid leukemia [286]. Each of the 12,362 samples consists of the measured response of a 2-drug pair tested on a patient’s cancer cells. The 1,235 input features are split between features describing the drug combinations and those describing the cancerous cells, which we modeled using the neural architectures described in [97] and [220].

According to the first-order explanations, the presence or absence of the drug Venetoclax in the drug combination was the most important feature. We also easily see that first-order explanations were inadequate in this case: while the presence of Venetoclax was generally predictive of a more responsive drug combination, the amount of positive response to this drug was predicted to vary across samples (see Fig. 3.6, top left).

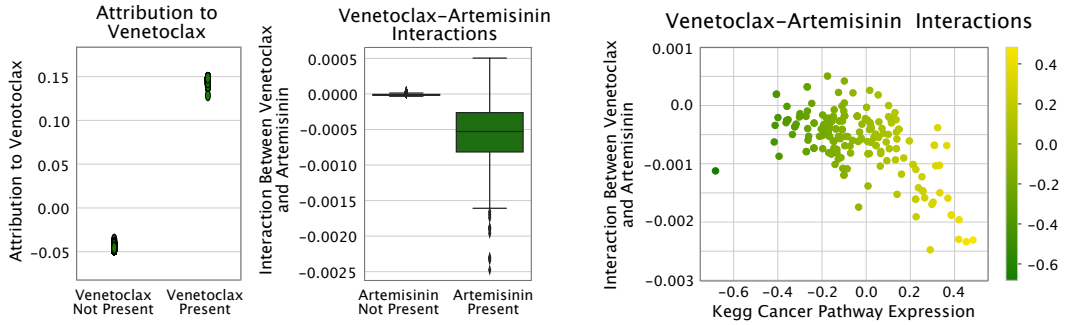

Figure 3.6: Left: Integrated Gradients values for Venetoclax. Middle: Venetoclax interactions with Artemisinin across all samples. Right: Venetoclax and Artemisinin interaction is driven by expression of genes in cancer samples.

Integrated Hessians reveals that some of this variability is attributable to the drug with which Venetoclax was combined. The model learned a strong negative interaction between Venetoclax and Artemisinin (see Fig. 3.6, middle), which we confirmed matched the ground truth ascertained from additional external data ($p = 2.31 \times 10^{-4}$). Finally, we gained insight into the variability in the *interaction* values between Venetoclax and Artemisinin by plotting them against the expression level of a pathway containing cancer genes (see Fig. 3.6, right). We found that patients with higher expression of this pathway tended to have a more negative interaction (sub-additive response) than those with a lower expression of it. Integrated Hessians enriches both our understanding of the interactions between drugs in our model and the genetic factors that influence this interaction.

3.7 Conclusion

We proposed a novel method called Integrated Hessians to explain feature interactions in neural networks. The interaction values we proposed have two natural interpretations: (1) as the effect of combining two features on a model's output, and (2) as the explanation of one feature's importance in terms of another. Our method provably satisfied common-sense axioms that previous methods did not and outperforms previous methods in practice at identifying known interactions on simulated data. Additionally, we demonstrated how to glean interactions from neural networks trained with a ReLU activation function that has no second derivative. In accordance with recent work, we showed why replacing the ReLU activation function with the SoftPlus at explanation time was both intuitive and efficient. Finally, we performed several experiments to reveal the utility of our method, from understanding performance gaps between model classes to discovering patterns a model has learned on

high-dimensional data.

We conclude that although feature attribution methods provide valuable insight into model behavior, such methods by no means end the discussion on interpretability. Rather, they encourage further work in deeper understanding model behavior. For example, our approach does not currently support the quantification of higher-order interactions between features. Future research to characterize such interactions would be valuable. For example, these methods may prove useful for image models, where individual pixels lack semantic meaning. Extending the efficiency or theoretical guarantees of existing methods that can detect these higher-order interactions, or developing new methods to provide these desiderata, may be of particular interest. Finally, while interactions add a degree of expressivity to feature attribution methods, finding approaches to explain models from outside the paradigm of feature attribution entirely would be creative and significant future research.

3.8 Supplement: Deriving Interaction Values

Here, we derive the formula for Integrated Hessians from its definition: $\Gamma_{i,j}(x) = \phi_j(\phi_i(x))$. We start by expanding ϕ_j using the definition of Integrated Gradients:

$$\Gamma_{i,j}(x) := (x_j - x'_j) \times \int_{\beta=0}^{1} \frac{\partial \phi_i(x' + \beta(x - x'))}{\partial x_j} d\beta. \tag{3.17}$$

We consider the function $\frac{\partial \phi_i}{\partial x_j}(x)$, and we first assume that $i \neq j$

$$\frac{\partial \phi_i}{\partial x_j}(x) = \tag{3.18}$$

$$(x_i - x'_i) \times \frac{\partial}{\partial x_j} \left(\int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha \right) = \tag{3.19}$$

$$(x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial}{\partial x_j} \left(\frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \right) d\alpha = \tag{3.20}$$

$$(x_i - x'_i) \times \int_{\alpha=0}^{1} \alpha \frac{\partial^2 f(x' + \alpha(x - x'))}{\partial x_i \partial x_j} d\alpha, \tag{3.21}$$

where we have assumed that the function f satisfies the conditions for the Leibniz Integral Rule (i.e., that integration and differentiation are interchangeable). These conditions require that the derivative of f, $\frac{\partial f}{\partial x_i}$ and its second derivative function $\frac{\partial^2 f}{\partial x_i \partial x_j}$ are continuous over x in the integration region, and that the bounds of integration are constant with respect to x. It is easy to see that the bounds of integration are constant with respect to x. It is also straightforward to see that common neural network activation functions — for example,

$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$, $\text{softplus}_\beta(x) = \frac{1}{\beta}\log(1+e^{-\beta x})$, or $\text{gelu}(x) = x\Phi(x)$, where $\Phi(x)$ is the cumulative distribution function of the normal distribution — have continuous first and second partial derivatives; this implies that compositions of these functions have continuous first and second partial derivatives, as well. Although this is not the case with the ReLU activation function, we discuss replacing it with SoftPlus in the main text.

We can proceed by plugging equation 3.21 into the original definition of $\Gamma_{i,j}(x)$:

$$\Gamma_{i,j}(x) := (x_j - x'_j) \times \int_{\beta=0}^{1} \frac{\partial \phi_i(x' + \beta(x - x'))}{\partial x_j} d\beta = \tag{3.22}$$

$$(x_j - x'_j) \times \int_{\beta=0}^{1} (x'_i - \beta(x_i - x'_i) - x'_i) \int_{\alpha=0}^{1} \alpha \frac{\partial^2 f(x' + \alpha(x' - \beta(x - x') - x'))}{\partial x_i \partial x_j} d\alpha d\beta = \tag{3.23}$$

$$(x_j - x'_j)(x_i - x'_i) \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \alpha\beta \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\alpha d\beta, \tag{3.24}$$

where all we've done is re-arrange terms.

Deriving $\Gamma_{i,i}(x)$ proceeds similarly:

$$\frac{\partial \phi_i}{\partial x_i}(x) = \tag{3.25}$$

$$\frac{\partial}{\partial x_i}\left((x_i - x'_i)\right) \times \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha + (x_i - x'_i) \times \frac{\partial}{\partial x_i}\left(\int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha\right) = \tag{3.26}$$

$$\int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha + (x_i - x'_i) \times \int_{\alpha=0}^{1} \alpha \frac{\partial^2 f(x' + \alpha(x - x'))}{\partial x_i \partial x_j} d\alpha, \tag{3.27}$$

using the chain rule. After similar re-arrangement, we arrive at

$$\Gamma_{i,i}(x) = (x_i - x'_i) \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha\beta(x - x'))}{\partial x_i} d\alpha d\beta + (x_i - x'_i)^2 \times \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \alpha\beta \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\alpha d\beta. \tag{3.28}$$

3.9 Supplement: Comparing Against Existing Methods

We now elaborate on the relationship between our method and six existing methods:

- Integrated Hessians: Our proposed method.

- Input Hessian: Uses the Hessian at the input instance. Using the input Hessian at a particular instance to measure interaction values is the natural generalization of using the gradient to measure the importance of individual features, as done by [258].

- Contextual Decomposition (CD): Introduced by [195] for LSTMs and extended to feed-forward and convolutional architectures by [260]. We focus on the generalized version introduced by the latter.

- Neural Interaction Detection (NID): Introduced by [283], this method generated interactions by inspecting the weights of the first layer of a feed-forward neural network.

- Shapley Interaction Index (SII): Introduced by [87] to allocate credit in coalitional game theory. It is used to explain interactions in neural networks similarly to how the Shapley value is used to explain attributions in [179].

- Shapley Taylor Interaction Index (STI): Introduced by [57], this method is similar to the Shapley Interaction Index, but it changes the weighting of certain coalitions in order to naturally fulfill the axiom that interactions should sum to the output.

- ArchDetect (AD): Introduced by [284], this method detects interactions between pairs of features using a quantity similar to a discrete second-order derivative.

- Group Expected Hessian (GEH): Introduced by [48], this method aggregates the input Hessian over many samples with respect to a Bayesian neural network.

- Deep Feature Interaction Maps (DFIM): Introduced by [89], this method determines interactions by seeing how much attributions change when features are perturbed.

We first discuss practical considerations regarding each method and then evaluate the degree to which each satisfies the axioms we identified in Section 2.2.

3.9.1 Comparison to Tsang et al. 2020

In Tables 1 and 2 of their recent paper, "How does this interaction affect me?: Interpretable attribution for feature interactions," [284] compare their "interaction attribution" method ArchAttribute to Integrated Hessians. We believe this is a faulty comparison. Their overall approach consists of three steps: (1) detect pairwise interactions between features using a method called ArchDetect, (2) use these pairwise interactions to cluster features into groups, such that interactions occur only between features within the same group, and (3) attribute importance to those groups of features using a method called ArchAttribute.

For a group $\mathcal{I}$ formed using the first two steps of this procedure, the ArchAttribute value is:

$$\phi(\mathcal{I}) = f(x^*_{\mathcal{I}} + x^{'}_{\setminus\mathcal{I}}) - f(x'), \tag{3.29}$$

where for p features in a data set, we let $\mathcal{I}$ be a subset of feature indices, $\mathcal{I} \subseteq \{1, 2, \cdots, p\}$. For a vector $x \in \mathbb{R}^p$, let $x_{\mathcal{I}} \in \mathbb{R}^p$ be defined as

$$(x_{\mathcal{I}})_i = \begin{cases} x_i, & \text{if } i \in \mathcal{I} \\ 0 & \text{otherwise.} \end{cases} \tag{3.30}$$

We let x^* represent the sample we want to explain, x' represent a neutral reference sample, and f be a black box model that takes a vector input and has a scalar output.

We prove why we believe this is a flawed comparison by showing that from a game-theoretic perspective, the value calculated by ArchAttribute corresponds to a Group Shapley Value rather than any sort of interaction value.

Theorem 2. *For a set of features $\mathcal{I}$ found using the first two steps of the procedure in [284], the ArchAttribute value $\Phi(\mathcal{I})$ is exactly the Group Shapley Value.*

Proof. Define a value function $v : 2^N \mapsto \mathbb{R}$ as $v(S) = f(x^*_S + x^{'}_{\setminus S})$, where x^* is the sample we want to explain and $x^{'}$ is the all zeros vector. Therefore, we can write:

$$\Phi(\mathcal{I}) = v(\mathcal{I}) - v(\emptyset). \tag{3.31}$$

Because the Archipelago method defines $\mathcal{I}$ as a set of features that do not interact with features outside of the set $\mathcal{I}$, we have that:

$$v(\mathcal{I}) - v(\emptyset) = v(\mathcal{I} \cup S) - v(\emptyset \cup S), \forall S \subseteq N \setminus \mathcal{I}. \tag{3.32}$$

For *any* family of coefficients $\{p^{\mathcal{I}}_S(N)\}_{S \subseteq N \setminus \mathcal{I}}$ that define a probability distribution on $2^{N \setminus \mathcal{I}}$, since these coefficients will sum to 1, we can say:

$$v(\mathcal{I} \cup S) - v(\emptyset \cup S) = \sum_{S \subseteq N \setminus \mathcal{I}} p^{\mathcal{I}}_S(N)[v(\mathcal{I} \cup S) - v(S)]. \tag{3.33}$$

The R.H.S. of the last equation defines a "probabilistic generalized value," the most well-known of which is the Group Shapley Value [187, 71].

□

If *values* measure the individual power of a player in a cooperative game, *generalized values* extend this notion to coalitions of players [187]. Since the ArchAttribute method appears to be a method for calculating *group attributions* rather than *feature interactions*, we instead compare to the interaction detection method ArchDetect, also described in [284]. The ArchDetect value for a pair of features i and j is given as

$$\bar{\omega}_{i,j} = \frac{1}{2}(\omega_{i,j}(x^*) + \omega_{i,j}(x^{'})), \tag{3.34}$$

where the quantities ω are defined:

$$\begin{aligned}\omega_{i,j}(x) = \big(\frac{1}{h_i h_j}\big(f(x^*_{\{i,j\}} + x_{\setminus\{i,j\}}) - f(x^{'}_{\{i\}} + x^*_{\{j\}} + x_{\setminus\{i,j\}}) \\ - f(x^*_{\{i\}} + x^{'}_{\{j\}} + x_{\setminus\{i,j\}}) + f(\boldsymbol{x}^{'}_{\{i,j\}} + x_{\setminus\{i,j\}})\big)\big)^2.\end{aligned} \tag{3.35}$$

If we substitute the more familiar set function notation, $v(S) = f(x^*_S + x^{'}_{\setminus S})$, we see that this value very closely resembles the Shapley Interaction Index [87], where only two contexts are considered and the terms within the summation are squared. Hence, it is a significantly better comparison to make with our method.

3.9.2 Practical Considerations

This section describes four desirable properties that interaction methods may or may not satisfy:

1. Local: A method is local if it operates at the level of individual predictions. It is global if it operates over the entire data set. Which is better is task dependent, but local methods are often more flexible because they can be aggregated globally [177].

2. Architecture Agnostic: A method is architecture agnostic if it can be applied to any neural network architecture.

3. Data Agnostic: A method is data agnostic if it can be applied to any type of data, no matter the structure.

4. Higher-Order Interactions: This paper primarily discusses interactions between pairs of features. However, some methods can generate interactions between groups of features larger than 2, and these are said to generate higher-order interactions.

Interaction Method	Local	Architecture Agnostic	Data Agnostic	Higher-Order Interactions
Input Hessian	✓	✓	✓	
Integrated Hessians (ours)	✓	✓	✓	
CD [260]	✓	∼*	✓	✓
NID [283]		✓	✓	✓
SII [87]	✓	✓	✓	✓
STI [57]	✓	✓	✓	✓
AD [284]	✓	✓	✓	
GEH [48]	✓		✓	
DFIM [89]	✓	✓		

Table 3.1: Comparing the practical properties of existing interaction methods. Because GEH and DFIM are neither architecture nor data agnostic, respectively, we omit them from empirical comparisons. *Contextual Decomposition was originally introduced for LSTMs in [195], and was generalized to feed-forward and convolutional architectures in [260]. However, the method has yet to be adapted to other architectures (e.g., transformers [56]).

In Table 3.1, we depict methods and the properties they satisfy. We note that GEH and DFIM are neither architecture nor data agnostic, respectively. Therefore, we do not include them in empirical comparisons since they cannot be run on feed-forward neural networks with arbitrary data. We also note that our method does not generate higher order interactions. Although in principle one could generate kth order interactions by recursively applying integrated gradients to itself k times, we do not discuss doing so in this paper.

3.9.3 Theoretical Considerations

We now evaluate which methods satisfy the axioms we presented in Section 2.2. Results are presented in Table 3.2. We note that Integrated Hessians and the Shapley Interaction Index share many theoretical properties, except that the Shapley Interaction Index trades completeness for the recursive axioms, which state that higher order interactions should be recursively defined from lower order ones. Whether this is preferable to satisfying completeness seems subjective.

The Shapley-Taylor Interaction Index satisfies completeness, although [57] refer to it as the "efficiency" axiom. They also introduce a new "interaction distribution" axiom, which guarantees that interaction effects between a set of featuers are not improperly credited towards any proper subset of the original set. However, their interaction index does not

provide a relationship between attribution values (Shapley values) and interaction values, while our method does. Which proprety is more desirable is, again, subjective.

The Hessian satisfies implementation invariance for the same reason that our method does: it relies only on the value of the network function and its derivatives. The hessian also satisfies (1) linearity since the derivative is a linear operator, and (2) symmetry preservation due to Schwarz's theorem. Contextual decomposition is symmetry preserving by definition but fails on implementation invariance due to the way it splits the bias.

In terms of computational efficiency, we can better appreciate the benefit of our approach compared to the Shapley Interaction Index and the Shapley Taylor interaction index by examining the asymptotic complexity of each algorithm. For a given neural network, forward and backward passes have the same asymptotic time complexity, which is a function of the number of nodes and layers (matrix multiplications) as well as the activation functions used [235]. Since the network itself does not change with the interaction method, we can therefore analyze each algorithm on the basis of the number of either forward or backward passes required to calculate all interactions. For each interaction, both the Shapley Interaction Index and the Shapley Taylor interaction index require considering an exponential number of subsets and require $O(2^d)$ forward passes. To calculate *all* interactions, these methods hence take $O(2^d d^2)$ forward passes. Integrated Hessians reduces the number of passes to be polynomial in the number of features, $O(kd^2)$, requiring k interpolation steps. Finally, heuristic methods like ArchDetect or Contextual Decomposition require $O(d^2)$ forward passes.

Although the Integrated Hessians method occupies an intermediate position in terms of asymptotic analysis, in practice we can calculate all possible pairs of interactions more quickly using this method due to the parallelization built into popular deep learning frameworks, back-propagating the Hessian through the model in parallel using GPUs [211, 2].

3.9.4 Quantitative Evaluations

We now elaborate on the quantitative comparisons presented in the main text. Note that we do not compare with GEH and DFIM because they are not applicable to feed-forward neural networks with continuous data.

Remove and Retrain

The Remove and Retrain benchmark starts with a trained network on a given data set. It ranks the features most important for prediction on every sample in the data set according to a given feature attribution method. Using the ranking, it iteratively ablates the most important features in each sample and then re-trains the model on the ablated data. To run this method, it is necessary to ablate a feature, which the original paper does by mean or zero

Interaction Method	Completeness	Interaction Sensitivity	Implementation Invariant
Input Hessian			✓
Integrated Hessians (ours)	✓	✓	✓
CD [260]		∼**	
NID [283]			
AD [284]		✓	✓
SII [87]		✓	✓
STI [57]	✓	✓	✓
GEH [48]			✓
DFIM [89]			∼***

Interaction Method	Symmetry Preserving	Interaction Linearity	Recursive Axioms*
Input Hessian	✓	✓	
Integrated Hessians (ours)	✓	✓	
CD [260]	✓		
NID [283]			
AD [284]	✓		
SII [87]	✓	✓	✓
STI [57]	✓	✓	
GEH [48]		✓	
DFIM [89]	∼***	∼***	

Table 3.2: Comparing the theoretical guarantees of interaction methods. We define each axiom in Section 2.2. *The recursive axioms are discussed in [87] and guarantee that higher-order interactions satisfy a specific recurrence relation. **Contextual Decomposition does not satisfy the exact sensitivity axiom we present, but it does guarantee that features equal to 0 will have zero interaction with any other feature. ***Whether or not Deep Feature Interaction Maps (DFIM) satisfies these properties relies on the underlying feature attribution method used: if the underlying method satisfies the properties, so does DFIM.

imputation [101]. However, doing so presents a problem when trying to compare methods that explain *interactions* between features rather than *attributions* to features.

Unlike ablating features, it is not straightforward to ablate an interaction between two features. Ablating both features in the interaction does not work, because it mixes main effects with interactions. Consider the function $f(x_1, x_2, x_3, x_4) = x_1 + x_2 + 0.1 * x_3 * x_4$. For $f(1.0, 1.0, 1.0, 1.0) = 2.1$, the largest and only interaction is between x_3 and x_4. However, ablating the pair x_1, x_2 would incur a larger performance hit because the features x_1 and x_2 have larger main effects than x_3 and x_4.

Instead, we opt to generate simulated data where the interactions are *known* and then ablate the interactions in the labels directly. We generate a simulated regression task with 10 features, where each feature is drawn independently from $\mathcal{N}(0, 1)$. The label is an additive sum of 20 interactions with random coefficients, drawn without replacement from all possible pairs of features. That is, for some specified interaction function g, we generate the label y as

$$\sum_{i=1}^{20} \alpha_i g(x_{i,1}, x_{i,2}),$$

where $x_{i,1}, x_{i,2}$ is the pair of features chosen to be part of the ith interaction and α_i are random coefficients drawn from a uniform distribution and normalized to sum to 1:

$$\sum_{i=1}^{20} \alpha_i = 1$$

.

To ablate an interaction, we multiply the interaction in the label with gaussian noise, which ensures ablating the largest interactions adds the most amount of noise to the label. For example, let $f(x_1, x_2, x_3) = 0.5 * g(x_1, x_2) + 0.3g(x_1, x_3)$. To ablate the interaction between x_1 and x_2, we compute the ablated label $\hat{f}(x_1, x_2, x_3) = \epsilon * 0.5 * g(x_1, x_2) + 0.3g(x_1, x_3)$, where $\epsilon \sim \mathcal{N}(0, 1)$. The interaction method that identifies the largest interactions in the data adds the largest amount of random noise into the label, thus increasing error the fastest.

For the simulated data set described in Section 5.2, we trained a 3 layer neural network with 64 hidden units each and ReLU activation. It achieved near-perfect performance on the simulated regression task, explaining over 99% of variance in the label. For each progressive ablation of an interaction, we retrained the network 5 times and re-evaluated performance. We then plotted the mean and standard deviation of performance on a held-out set, as recommended by the original paper [101]. In Section 5.5, we showed the results for $g_{\text{tanhsum}}(x_i, x_j) = \tanh(x_i + x_j)$. We show results for these four additional interaction types:

- $g_{\text{cossum}}(x_i, x_j) = \cos(x_i + x_j)$
- $g_{\text{multiply}}(x_i, x_j) = x_i * x_j$

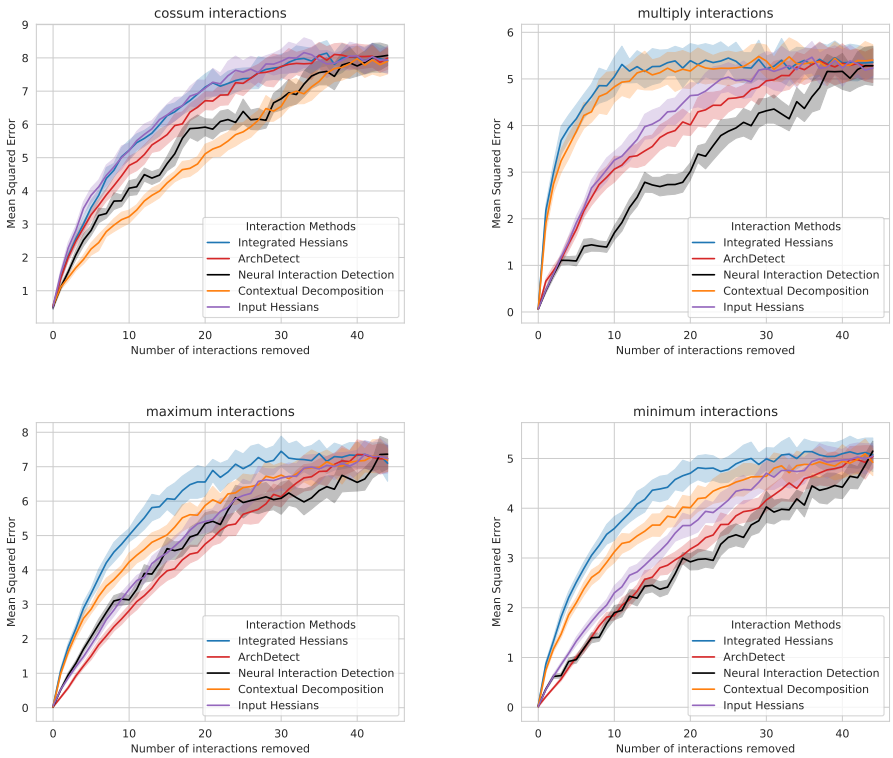

Figure 3.7: Additional comparisons with heuristic methods on known simulated interactions. The charts plot the mean and standard deviation across 5 re-trainings of the model as a function of ablated interactions. Our method (in blue) outperforms all existing heuristic methods on all interaction types.

- $g_{\text{maximum}}(x_i, x_j) = \max(x_i, x_j)$
- $g_{\text{minimum}}(x_i, x_j) = \min(x_i, x_j)$

The results, in Figure 3.7 and Figure 3.8, show that our method consistently outperforms all other methods except the Monte Carlo estimation of the Shapley Interaction Index, which performs as well as our method. As discussed in Section 5.1, our method is much more computationally tractable than the Shapley Interaction Index, even using Monte Carlo estimation. We also observe that Neural Interaction Detection [283] is not a local interaction method; rather, it detects interactions globally. To compare with it, we simply ablate the top-ranked interaction globally in all samples.

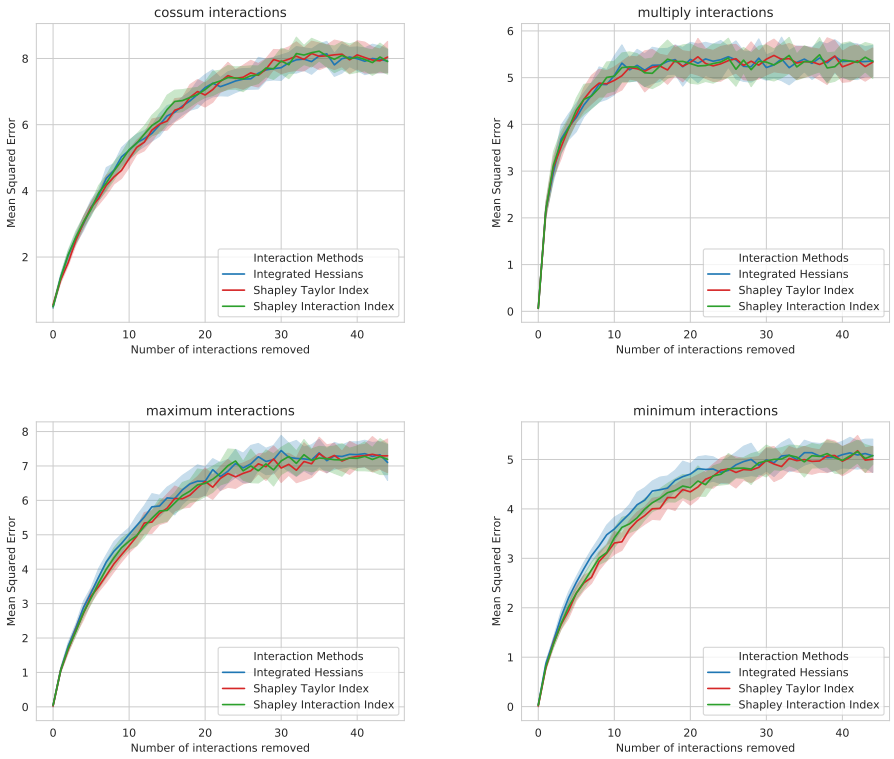

Figure 3.8: Additional comparisons with Shapley value-based methods on known simulated interactions (Top Left: cossum, Top Right: multiplicative, Bottom Left: maximum, Bottom Right: minimum). The charts plot the mean and standard deviation across 5 re-trainings of the model as a function of ablated interactions. Our method (in blue) performs comparably to the Shapley value-based methods on all interaction types but is significantly more computationally efficient.

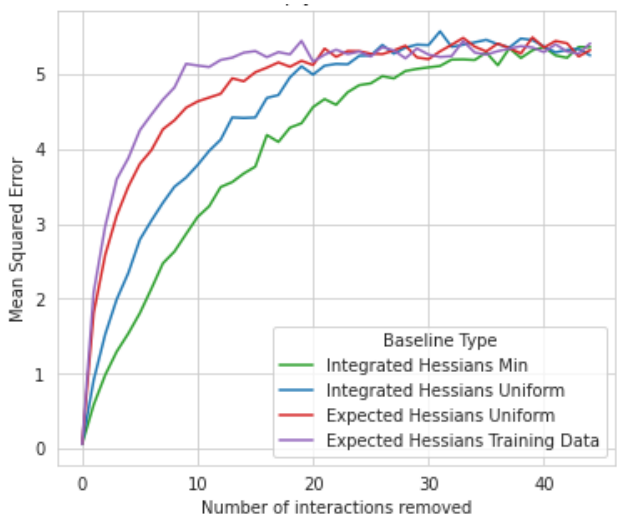

Figure 3.9: Using one of our synthetic data sets (with pairwise multiplicative interaction label), we demonstrate how for certain tasks, the choice of baseline can impact the quality of interaction detection. Both methods that take an expectation over distributions of baselines outperform single baseline methods, while using the training set as the distribution of baselines outperforms sampling baselines from the product of uniform distributions over the range of each individual feature.

Impact of baseline

To demonstrate the impact of baseline selection on interaction detection, we re-ran our benchmark for interaction detection using four different choices of baseline. The first single baseline, called "min," sets each element in the baseline vector to its minimum observed value from the training data. It is similar to using the black image in image attribution. The second, "uniform," is a single randomly selected baseline from the product of uniform distributions ranging from the minimum to maximum observed values of each feature for the training data. The third baseline is a random sample (expected hessians) over baselines drawn from the product of uniform distributions mentioned above. The fourth is a random sample over baselines drawn from the training distribution (the approach we defined in Section 2.1). In Figure 3.9 we note that (1) both approaches that took an expectation over multiple baselines outperformed the single baseline options, and (2) that sampling baselines from the training data outperformed sampling baselines from the product of uniform distributions covering the feature ranges.

Sanity Checks

To ensure that our interaction attributions were sensitive to network and data randomization, we tested our approach using the "sanity checks" proposed in [3]. For this experiment, we used a data set of synthetic features, which consisted of five 0-mean unit-variance independent Gaussian random variables. The synthetic label was the sum of each of the ten possible pairwise multiplicative interactions between features with coefficients ranging in magnitude from 10 to 1. For our model, we used a neural network with two hidden layers and Tanh non-linearities until convergence (validation set $> R^2 : 0.99$). We next fit a network and found the interactions using Integrated Hessians. We then compared the rank correlation of these interactions with the interactions attained from explaining (1) a network with randomly initialized weights and (2) a network trained to convergence on the training set on data where labels were shuffled at random (but the features were the same). In *both* settings, we found that the Spearman correlation between the true and randomized interactions was 0. This indicates that our method passed the sanity checks in [3].

3.10 Supplement: Effects of Smoothing ReLU Networks

3.10.1 Proof of Theorem 1

Theorem 1 For a one-layer neural network with softplus_β non-linearity, $f_\beta(x) = \text{softplus}_\beta(w^T x)$, and d input features, we can bound the number of interpolation points k needed to approximate the Integrated Hessians to a given error tolerance ϵ by $k \leq \mathcal{O}(\frac{d\beta^2}{\epsilon})$.

Proof. As pointed out in [274] and [268], completeness can be used to assess the convergence of the approximation. We first show that decreasing β improves convergence for Integrated Gradients. To accurately calculate the Integrated Gradients value Φ for a feature i, we need to bound the error between the approximate computation and exact value. The exact value is given as:

$$\Phi_i(\theta, x, x') = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f(x'(1-\alpha) + x\alpha)}{\partial x_i} d\alpha. \tag{3.36}$$

To simplify notation, we can define the partial derivative that we want to integrate over in the ith coordinate as $g_i(x) = \frac{\partial F(x)}{\partial x_i}$:

$$\Phi_i(\theta, x, x') = (x_i - x_i') \times \int_{\alpha=0}^{1} g_i(x'\alpha + x(1-\alpha)) d\alpha. \tag{3.37}$$

Since the single-layer neural network with SoftPlus activation is monotonic along the path, the error in the approximate integral can be lower bounded by the left Riemann sum L_k:

$$L_k = \frac{\|x - x'\|}{k} \sum_{i=0}^{k-1} g_i(x' + \frac{i}{k}(x - x')) \leq \int_{\alpha=0}^{1} g_i(x'\alpha + x(1-\alpha))d\alpha \tag{3.38}$$

and can likewise be upper-bounded by the right Riemann sum R_k:

$$\int_{\alpha=0}^{1} g_i(x'\alpha + x(1-\alpha))d\alpha \leq R_k = \frac{\|x - x'\|}{k} \sum_{i=1}^{k} g_i(x' + \frac{i}{k}(x - x')). \tag{3.39}$$

We can then bound the magnitude of the error between the Riemann sum and the true integral by the difference between the right and left sums:

$$\epsilon \leq |R_k - L_k| = \frac{\|x - x'\|_2}{k} |g_i(x) - g_i(x')|. \tag{3.40}$$

By the mean value theorem, we know that for some $\eta \in [0, 1]$ and $z = x' + \eta(x - x')$, $g_i(x) - g_i(x') = \nabla_x g_i(z)^\top (x - x')$. Therefore:

$$\epsilon \leq \frac{\|x - x'\|}{k} \nabla_x g_i(z)^\top (x - x'). \tag{3.41}$$

Rewriting in terms of the original function, we have:

$$\epsilon \leq \frac{\|x - x'\|}{k} \sum_{j=0}^{d} \left(\frac{\partial^2 f(z)}{\partial x_i x_j} (x_j - x'_j) \right). \tag{3.42}$$

We can then consider the gradient vector of $g_i(x)$:

$$\nabla_x g_i(x) = \left[\frac{\beta w_0 e^{\beta w^T x}}{(e^{\beta w^T x} + 1)^2}, \frac{\beta w_1 e^{\beta w^T x}}{(e^{\beta w^T x} + 1)^2}, \cdots \right] \tag{3.43}$$

where each coordinate is maximized at the zeros input vector and takes a maximum value of $\beta w_i / 4$. We can therefore bound the error in convergence as:

$$\epsilon \leq \frac{\|x - x'\|}{k} \sum_{j=0}^{d} \left(\frac{\beta \|w\|_\infty}{4} (x_j - x'_j) \right). \tag{3.44}$$

Ignoring the dependency on path length and the magnitude of the weights of the neural network, we see that:

$$k \leq \mathcal{O}\left(\frac{d\beta}{\epsilon} \right). \tag{3.45}$$

This demonstrates that the number of interpolation points k necessary to achieve a set error rate ϵ decreases as the activation function is smoothed (the value of β decreases). While this

proof bounds the error in the approximation of the integral for a single feature, we get the error in completeness by multiplying by an additional factor of d features.

We can extend the same proof to Integrated Hessians values. We first consider the error for estimating off-diagonal terms $\Gamma_{i,j}, i \neq j$. The true value we are trying to approximate is given as:

$$\Gamma_{ij} = (x_i - x'_i)(x_j - x'_j) \times \int_{\alpha\beta} \frac{\partial^2 f(x' + \alpha\beta(x - x'))}{\partial x_i \partial x_j} d\alpha d\beta. \tag{3.46}$$

For simplicity of notation, we can say $h_{ij}(x) = \frac{\partial^2 F(x)}{\partial x_i \partial x_j}$. Assuming that we are integrating from the all-zeros baseline (as suggested in [274]), since $h_{ij}(x) =$ is monotonic on either interval from the 0 baseline, we can again bound the error in the double integral by the magnitude of the difference in the left and right Riemann sums:

$$\epsilon \leq \left| \frac{\|x - x'\|_2^2}{k} \sum_{j=1}^{k} \sum_{i=1}^{k} h_{ij}(x' + \frac{ij}{k}(x - x')) - \frac{\|x - x'\|_2^2}{k^2} \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} h_{ij}(x' + \frac{ij}{k}(x - x')) \right|, \tag{3.47}$$

$$\epsilon \leq \frac{\|x - x'\|_2^2}{k} \left| \big(h_{ij}(x) + 2 \sum_{i=1}^{k-1} h_{ij}(x' + \frac{i}{\sqrt{k}}(x - x'))\big) - \big(h_{ij}(x') + 2 \sum_{i=1}^{k-1} h_{ij}(x')\big) \right|. \tag{3.48}$$

We can then use monotonicity over the interval to say that $h_{ij}(x' + \frac{i}{\sqrt{k}}(x - x')) < h_{ij}(x)$, which gives us:

$$\epsilon \leq \frac{(2k-1)\|x - x'\|_2^2}{k} \left| h_{ij}(x) - h_{ij}(x') \right|. \tag{3.49}$$

By the mean value theorem, we know that for some $\beta \in [0, 1]$, $h_{ij}(x) - h_{ij}(x') = \nabla_x h_{ij}(\beta)^\top (x - x')$. Substituting gives us:

$$\epsilon \leq \frac{(2k-1)\|x - x'\|_2^2}{k} \nabla_x h_{ij}(\beta)^\top (x - x'). \tag{3.50}$$

We can then consider the elements of the gradient vector:

$$\nabla_x h_{ij}(x) = \left[- \frac{\beta^2 w_i w_j w_1 e^{\beta w^T x} \big(e^{\beta w^T x} - 1\big)}{(e^{\beta w^T x} + 1)^3}, - \frac{\beta^2 w_i w_j w_2 e^{\beta w^T x} \big(e^{\beta w^T x} - 1\big)}{(e^{\beta w^T x} + 1)^3}, \cdots \right]. \tag{3.51}$$

For the univariate version of each coordinate, we can maximize the function by taking the derivative with respect to x and setting it equal to 0:

$$\frac{d}{dx} \left(- \frac{\beta^2 e^{\beta x} \big(e^{\beta x} - 1\big)}{(e^{\beta x} + 1)^3} \right) = \frac{\beta^3 e^{\beta x} (-4e^{\beta x} + e^{2\beta x} + 1)}{(e^{\beta x} + 1)^4} = 0. \tag{3.52}$$

We can see that this equation holds only when $(-4e^{\beta x} + e^{2\beta x} + 1) = 0$, and we can solve it by finding the roots of this quadratic equation, which occur when $x = \frac{1}{\beta}\log(2 \pm \sqrt{3})$. When we plug that back in, we find the absolute value of the function in that coordinate takes a maximum value of $\frac{\beta^2}{6\sqrt{3}}$. Therefore, for a given set of fixed network weights, we observe that the coordinate-wise maximum magnitude of $\nabla_x h_{ij}, \propto \beta^2$ and that the number of interpolation points necessary to reach a desired level of error in approximating the double integral decreases as β is decreased. Again ignoring the fixed weights and path length, the number of interpolation points necessary is bounded by

$$k \leq \mathcal{O}\left(\frac{d\beta^2}{k}\right). \tag{3.53}$$

For the $i = j$ terms (main effect terms), the error will have another additive factor of β, since main effect has an additional term equal to:

$$(x_i - x_i') \int_{\beta=0}^{1} \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha\beta(x - x'))}{\partial x_i} d\alpha d\beta. \tag{3.54}$$

When we bound the error in this approximate integral by the difference between the double left sum and double right sum, we find that:

$$\epsilon \leq \frac{(2k-1)\|x - x'\|_2^2}{k} \big| g_i(x) - g_i(x') \big|. \tag{3.55}$$

Following the exact same steps as in 3.40 through 3.44, we can then show the bound on the error of the on-diagonal terms will have an additional term that is $\propto \beta$. Due to the axiom of interaction completeness, the error bound of the entire convergence can be obtained by summing all individual terms, incurring another factor of d^2 in the bound.

□

3.10.2 SoftPlus Activation Empirically Improves Convergence

In addition to theoretically analyzing the effects of smoothing the activation functions of a single-layer neural network on the convergence of the approximate values of Integrated Gradients and Integrated Hessians, we wanted to empirically analyze the same phenomenon in deeper networks. We first created two networks: one with 5 hidden layers of 50 nodes, and a second with 10 hidden layers of 50 nodes. We then randomly initialized these networks using the Xavier Uniform intialization scheme [85]. We created 10 samples to explain, each with 100 features drawn at random from the standard normal distribution. To evaluate the convergence of our approximate Integrated Hessians values, we plotted the interaction completeness error (the difference between model output and the sum of Integrated Hessians

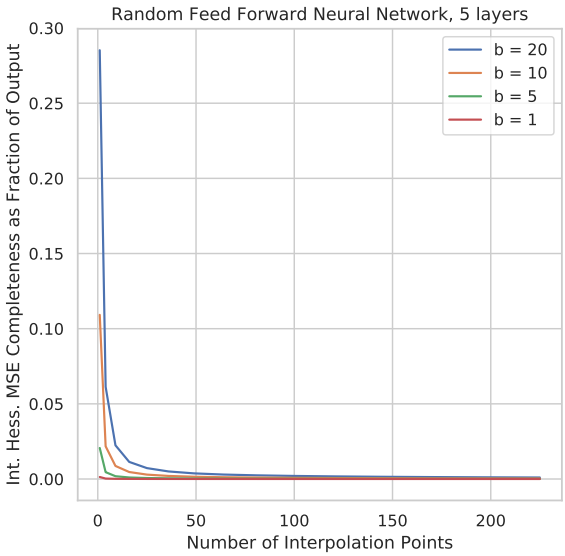

Figure 3.10: 5-Layer Network Results. Interaction completeness error (difference between model output and sum of Integrated Hessians values) decreases more quickly with the number of interpolation points as the β parameter for the SoftPlus activation function is decreased (as the function is smoothed). Results are averaged over 10 samples with 100 features for a neural network with 5 hidden layers of 50 nodes each.

values) as a fraction of the magnitude of the function output. As we decreased the value of β, we smoothed the activations. We observed that the number of interpolations required to converge decreased (see Fig. 3.10 and Fig. 3.11). Note that the randomly initialized weights of each network were held constant, and only the value of β in the activation function changed.

3.11 Supplement: Details on the Sentiment Analysis Task

3.11.1 Fine-Tuning DistilBERT

As mentioned in Section 6.1, we downloaded pre-trained weights for DistilBERT, a pre-trained language model introduced in [242], from the HuggingFace Transformers library [297]. We fine-tuned the model on the Stanford Sentiment Treebank data set introduced by [263]. We fine-tuned for 3 epochs using a batch size of 32 and a learning rate of 0.00003. We used a a max sequence length of 128 tokens, and the Adam algorithm for optimization [131]. We tokenize

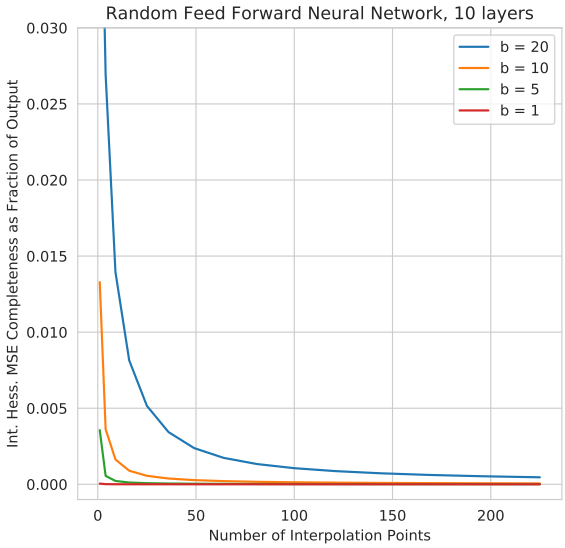

Figure 3.11: 10-Layer Network Results. Decreasing β has an even more dramatic effect on convergence when the network is deeper than in the 5-layer case. Again, this result shows that the interaction completeness error decreases more quickly with the number of interpolation points as the activation function is smoothed. Results are averaged over 10 samples with 100 features for a neural network with 10 hidden layers of 50 nodes each.

using the HuggingFace uncased tokenizer. We did not search for these hyper-parameters; rather, they were the defaults presented for fine-tuning in the HuggingFace repository. We found that they worked adequately for our purposes, so we did not attempt to search through additional hyperparameters.

3.11.2 Training a CNN

The convolutional neural network we used for comparison in Section 6.1 was trained from scratch on the same data set. We randomly initialized 32-dimensional embeddings and used a max sequence length of 52. First, we applied dropout to the embeddings, with a dropout rate of 0.5. The network itself was composed of 1D convolutions with 32 filters of size 3 and 32 filters of size 8. Each filter size was applied separately to the embedding layer, after which max pooling with a stride of 2 was applied; the output of both convolutions was then concatenated and fed through a dropout layer, with a dropout rate of 0.5 during training. A hidden layer of size 50 followed the dropout, finally followed by a linear layer generating a scalar prediction to which the sigmoid function was applied.

We trained with a batch size of 128 for 2 epochs and used a learning rate of 0.001. We optimized using the Adam algorithm with the default hyper-parameters [131]. Since this model was not pre-trained on a large language corpus and lacks the expressive power of a deep transformer, it cannot capture patterns like negation that a fine-tuned DistilBERT can.

3.11.3 Generating Attributions and Interactions

To generate attributions and interactions, we used Integrated Gradients and Integrated Hessians with the zero-embedding baselinne, i.e., the embedding produced by the all zeros vector, which normally encodes the padding token. Because embedding layers are not differentiable, we generated attributions and interactions to the word embeddings and then summed over the embedding dimension to get word-level attributions and interactions, as done in [274]. When computing attributions and interactions, we used 256 background samples. Because DistilBERT uses the GeLU activation function [226], which has continuous first and second partial derivatives, there was no need to use the SoftPlus replacement. When we plotted interactions, we avoided plotting the main-effect terms in order to better visualize the interactions between words.

3.11.4 Additional Examples of Interactions

Here, we include additional examples of interactions learned on the sentiment analysis task. First, we expand upon the idea of saturation in natural language, displayed in Figure 3.12. We display interactions learned by a fine-tuned DistilBERT on the following phrases: “a bad movie”

(negative with 0.9981 confidence), "a bad, terrible movie" (negative with 0.9983 confidence), "a bad, terrible, awful movie" (negative with 0.9984 confidence), and "a bad, terrible, awful, horrible movie" (negative with 0.9984 confidence). The confidence of the network saturates: a network output can get only so negative before it begins to flatten. However, the number of negative adjectives in the sentence increases. This means a sensible network would spread the same amount of credit (because the attributions sum to the saturated output) across a larger number of negative words, which is exactly what DistilBERT does. However, each word then gets less negative attribution than it would if it were alone. Thus, negative words have positive interaction effects, which is exactly what we see from the figure.

In Figure 3.13, we show another example of the full interaction matrix on a sentence from the validation set. In Figure 3.14, we present an example of how explaining the importance of a particular word can indicate whether that word is important because of its main effect or because of its surrounding context. We show additional examples from the validation set in Figures 3.15, 3.16, 3.17, 3.18, 3.19. Note that while some interactions make intuitive sense to humans ("better suited" being negative or "good script" being positive), many others are less intuitive. These interactions could indicate that the Stanford Sentiment Treebank data set does not fully capture the expressive power of language (e.g., it does not have enough samples to fully represent all possible interactions in language), or they could indicate that the model has learned higher order effects that cannot be explained by pairwise interactions alone.

3.12 Supplement: Additional Experiments

3.12.1 Heart Disease Prediction

We aggregated interactions learned from many samples in a clinical data set and used them to reveal global patterns. We examined the Cleveland heart disease data set [54, 49]. After preprocessing, the data set contained 298 patients with 13 associated features, including demographic information like age and gender and clinical measurements such as systolic blood pressure and serum cholesterol. The task was to predict whether a patient had coronary artery disease. The list of features, which we reproduce here, is from [54], the original paper introducing the data set:

1. Age of patient (mean: 54.5 years ± standard deviation: 9.0)

2. Gender (202 male, 96 female)

3. Resting systolic blood pressure (131.6 mm Hg ± 17.7)

4. Cholesterol (246.9 mg/dl ± 51.9)

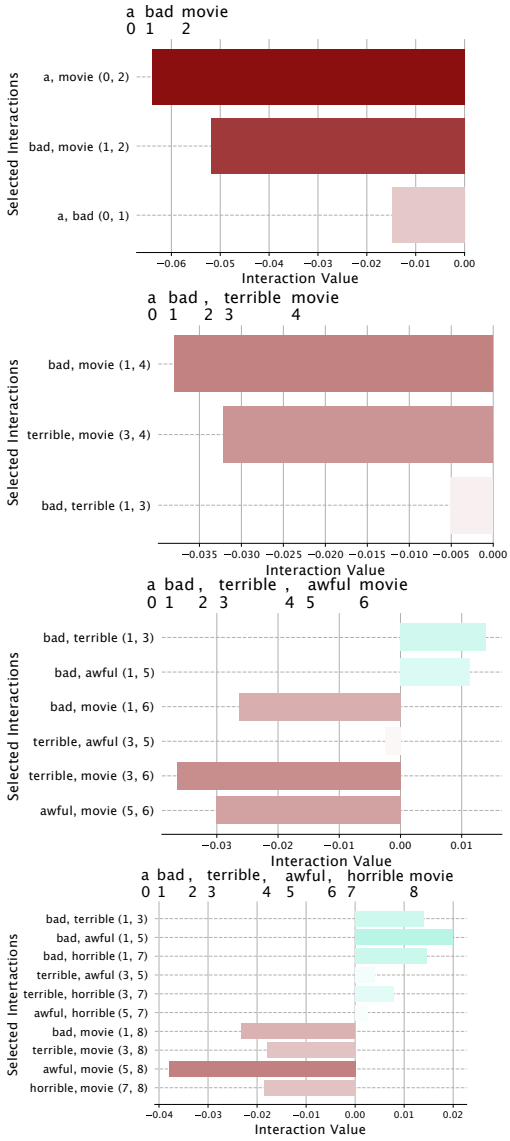

Figure 3.12: The effects of increasing saturation. As we add more negative adjectives to describe the word "movie," they interact increasingly positively even though they interact negatively with the word they describe. This is because each individual negative adjective has less impact on the overall negativity of the sentence the more negative adjectives there are.

Figure 3.13: An example from the Stanford Sentiment Analysis Treebank validation set. Interactions highlight intuitive patterns in text, such as phrases like “beautifully observed" and “miraculously unsentimental" being strongly positive interactions.

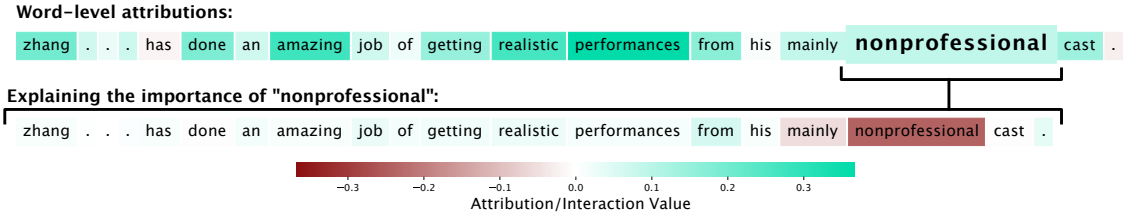

Figure 3.14: An example from the Stanford Sentiment Analysis Treebank validation set. This example shows that the word “nonprofessional" has a main effect that is negative, but the surrounding context outweighs the main effect and makes the overall attribution positive.

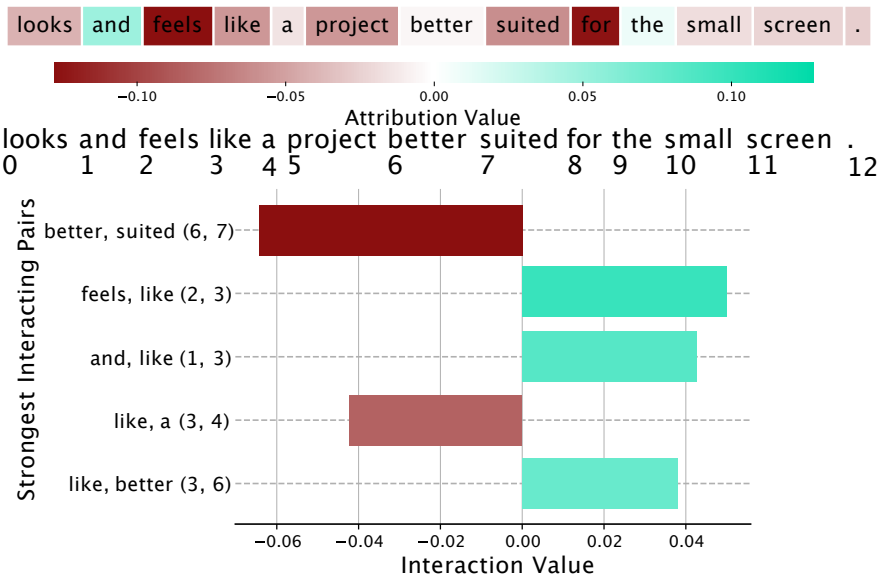

Figure 3.15: An example from the Stanford Sentiment Analysis Treebank validation set. This example highlights the phrase "better suited" being strongly negative, which is very intuitive. However, some of the other interactions are slightly less intuitive and may indicate lack of training data or higher order interactions beyond word pairs.

5. Whether or not a patient's fasting blood sugar was above 120 mg/dl (44 yes)

6. Maximum heart rate achieved by exercise (149.5 bpm ± 23.0)

7. Whether or not a patient has exercise-induced angina (98 yes)

8. Excercise-induced ST-segment depression (1.05 mm ± 1.16)

9. Number of major vessels appearing to contain calcium as revealed by cinefluoroscopy (175 patients with 0, 65 with 1, 38 with 2, 20 with 3)

10. Type of pain a patient experienced if any (49 experienced typical anginal pain, 84 experienced atypical anginal pain, 23 experienced non-anginal pain and 142 patients experienced no chest pain)

11. Slope of peak exercise ST segment (21 patients had upsloping segments, 138 had flat segments, 139 had downsloping segments)

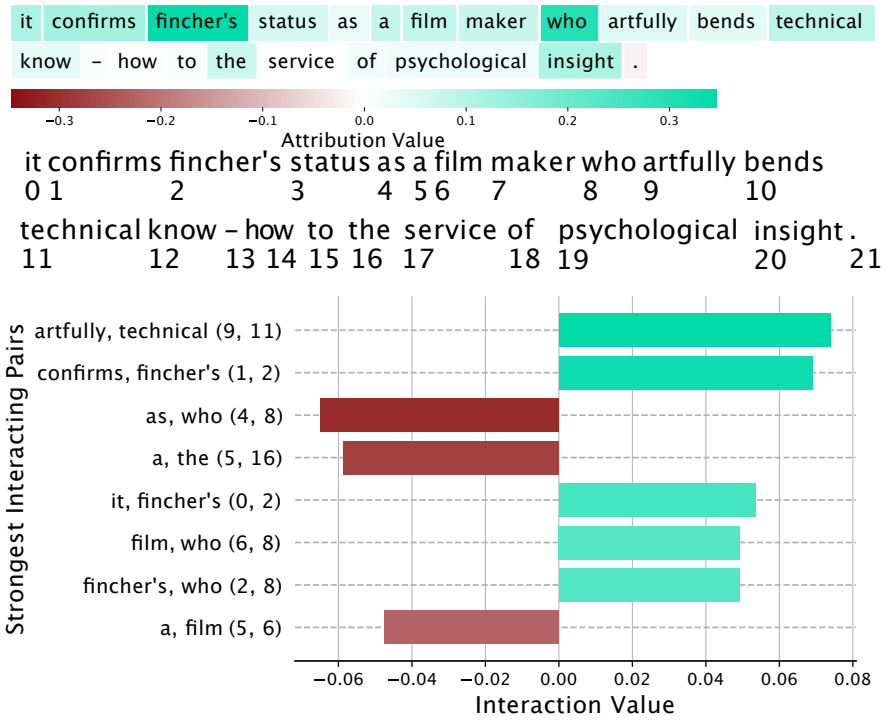

Figure 3.16: Another example from the Stanford Sentiment Analysis Treebank validation set. Notice how the strongest interact pairs may not necessarily be adjacent words, e.g., "artfully" and "technical."

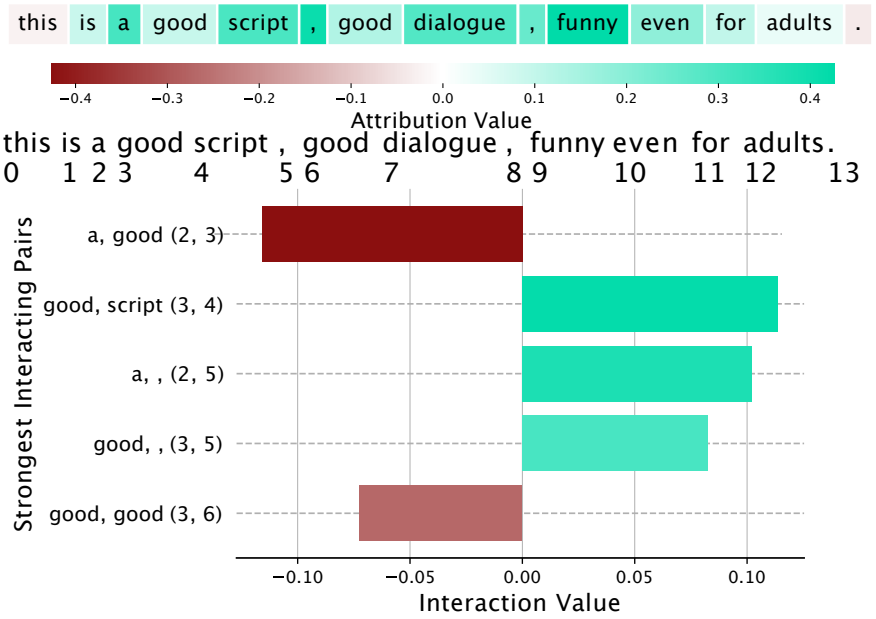

Figure 3.17: An example from the Stanford Sentiment Analysis Treebank validation set. Interestingly, the phrase "a good" has a negative interaction. This may indicate saturation effects, higher order effects, or that the model has simply learned an unintuitive pattern.

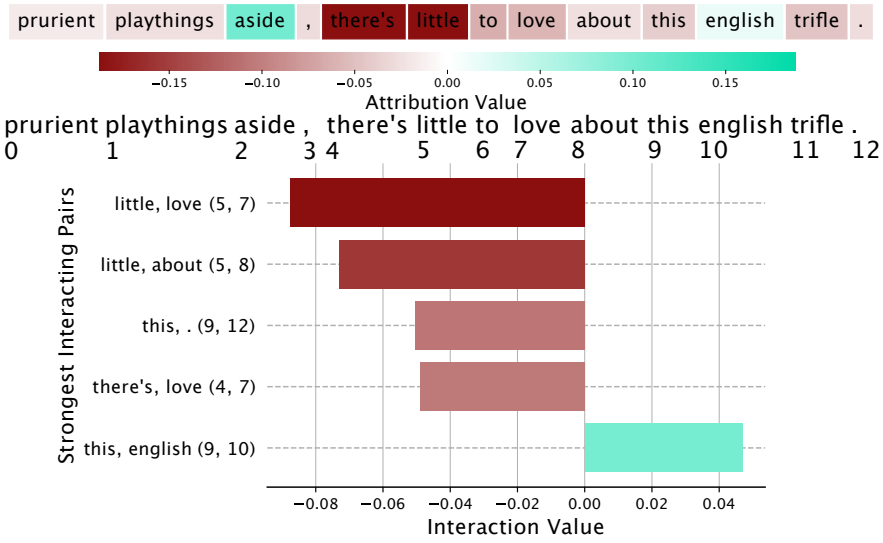

Figure 3.18: An example from the Stanford Sentiment Analysis Treebank validation set. This example shows some very intuitive negative interactions among the phrase "there's little to love". Interestingly, "this english" has a positive interaction: perhaps the data set has a bias for English movies?

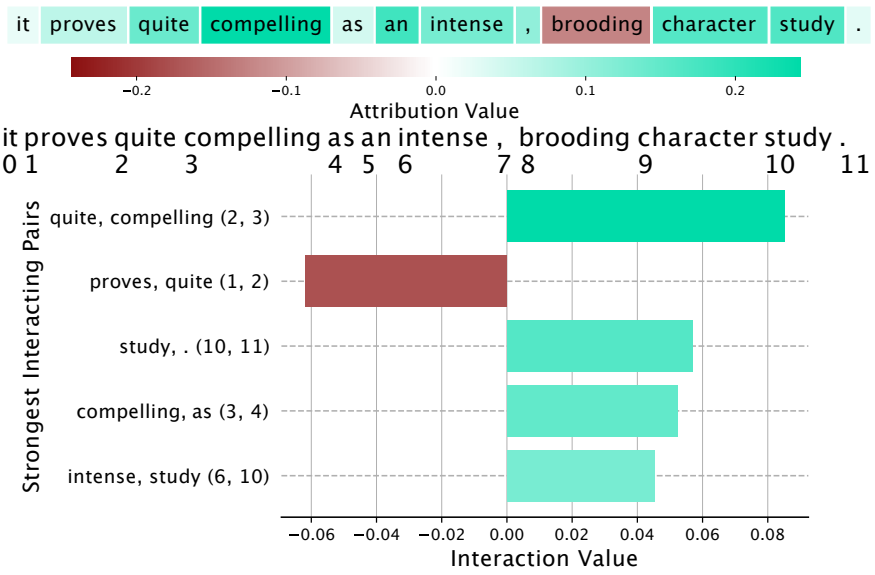

Figure 3.19: An example from the Stanford Sentiment Analysis Treebank validation set. This also shows intuitive patterns, e.g., "quite compelling" being strongly positive.

12. Whether or not a patient had thallium defects as revealed by scintigraphy (2 patients with no information available, 18 with fixed defects, 115 with reversible defects and 163 with no defects)

13. Classification of resting electrocardiogram (146 with normal resting ecg, 148 with an ST-T wave abnormality, and 4 with probable or definite left centricular hypertrophy)

We split the data into 238 patients for training (of which 109 had coronary artery disease) and 60 for testing (of which 28 have coronary artery disease). We used a two-layer neural network with 128 and 64 hidden units, respectively, with SoftPlus activation after each layer. We optimized using gradient descent (processing the entire training set in a single batch) with an initial learning rate of 0.1 that decays exponentially with a rate 0.99 after each epoch. We used nesterov momentum with $\beta = 0.9$ [276]. After training for 200 epochs, the network achieved a held-out accuracy of 0.8667, with a 0.8214 true positive rate and a 0.9062 true negative rate. Note that the hyper-parameters chosen here were not carefully tuned on a validation set - they simply seemed to converge to a reasonable performance on the training set. Our focus is not making state-of-the-art predictions or comparing model performance, but rather interpreting the patterns a reasonable model learns.

To generate attributions and interactions for this data set, we used Expected Gradients and Expected Hessians, with the training set forming the background distribution. We used

200 samples to compute both attributions and interactions, although we note this number is probably larger than necessary but was easy to compute due to the data set's small size.

Figure 3.20 shows which features were most important for predicting heart disease aggregated over the entire data set, as well as the trend of importance values. Interestingly, the model learns some strangely unintuitive trends: if a patient did not experience chest pain, they were more likely to have heart disease than if they experienced anginal chest pain. This could indicate problems with the way certain features were encoded, or perhaps data set bias. Figure 3.21 demonstrates an interaction learned by the network between maximum heart rate achieved and gender, and Figure 3.22 demonstrates an interaction between exercise-induced ST-segment depression and the number of major vessels appearing to contain calcium.

In Figure 3.23, we examine interactions with a feature describing the number of major coronary arteries with calcium accumulation (0 to 3), as determined by cardiac cinefluoroscopy [55]. Previous research has shown that this technique is a reliable way to gauge calcium build-up in major blood vessels, and it serves as a strong predictor of coronary artery disease [55, 15, 169]. Our model correctly learned that more coronary arteries with evidence of calcification indicate increased risk of disease. Additionally, Integrated Hessians reveals that our model learns a negative interaction between the number of coronary arteries with calcium accumulation and female gender. This supports the well-known phenomenon of under-recognition of heart disease in women – at the same levels of cardiac risk factors, women are less likely to have clinically manifest coronary artery disease [182].

3.12.2 Pulsar Star Prediction

We used a physics data set to confirm that a model learned a global pattern that was visible in the training data. We utilized the HRTU2 data set, curated by [181] and originally gathered by [123]. The task was to predict whether or not a particular signal measured from a radio telescope was a pulsar star or generated from radio frequency interference (e.g. background noise). The features included statistical descriptors of measurements made from the radio telescope. The data set contained 16,259 examples generated through radio frequency interference and 1,639 examples that were pulsars. The data set had 4 statistical descriptors — mean, standard deviation, skewness and kurtosis — of two measurements relating to pulsar stars: the *integrated pulse profile* (IP) and the *dispersion-measure signal-to-noise ratio curve* (DM-SNR), for a total of 8 features. The integrated pulse profile measures how much signal the supposed pulsar star emits as a function of the phase of the pulsar: as pulsars rotate, they emit radiation from their magnetic poles, which periodically sweeps over the earth. We can measure the radiation over time using a radio telescope and aggregating measurements over the phase to get the IP. Signals that are pulsar stars should in theory have stronger, more peaked integrated pulse profiles than those generated from radio frequency

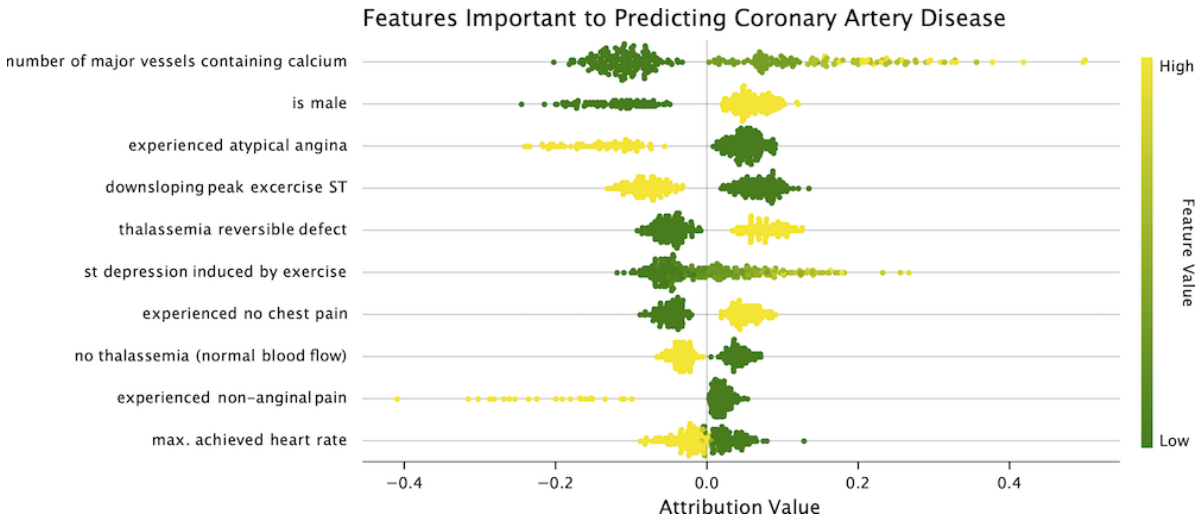

Figure 3.20: A summary of the most important features for predicting heart disease. A positive attribution indicates increased risk of heart disease (negative value indicates decreased risk of heart disease). The features are ordered by largest mean absolute magnitude over the data set. For binary features, high (yellow) indicates true while low (green) indicates false. For example, for the feature "experienced atypical angina," yellow means the patient did experience atypical angina, and green means the patient did not.

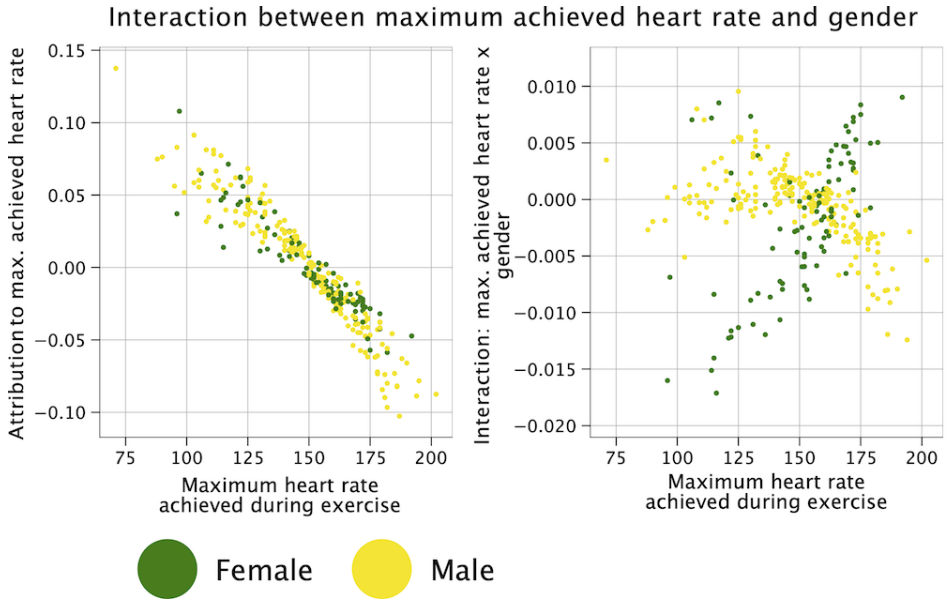

Figure 3.21: An interaction learned by the model between maximum achieved heart rate during exercise and the gender of the patient. In general, achieving a higher heart rate during exercise indicated a lower risk of heart disease, but the model learns that this pattern is stronger for men than for women.

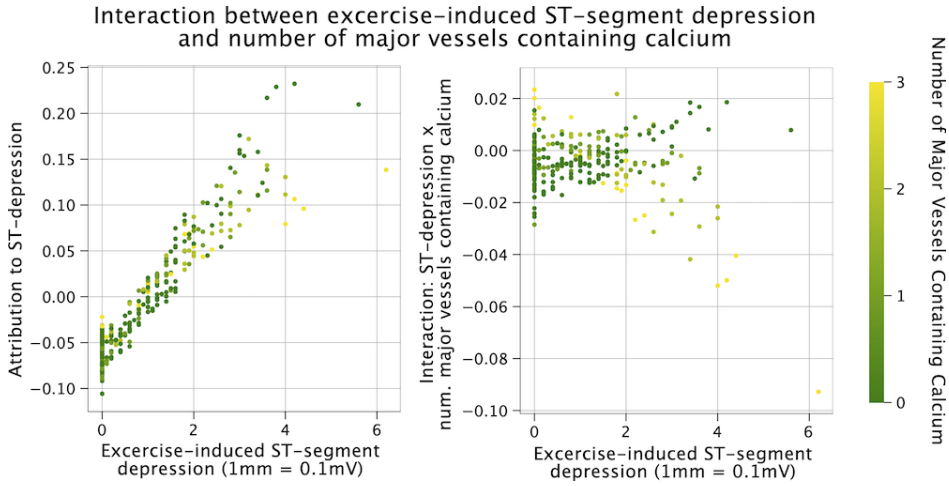

Figure 3.22: An interaction learned between ST-segment depression and the number of major vessels appearing to contain calcium. The interaction seems to indicate that if a patient has many vessels appearing to contain calcium, then st-segment depression is less important toward driving risk, probably because the number of major vessels containing calcium becomes the main risk driver.

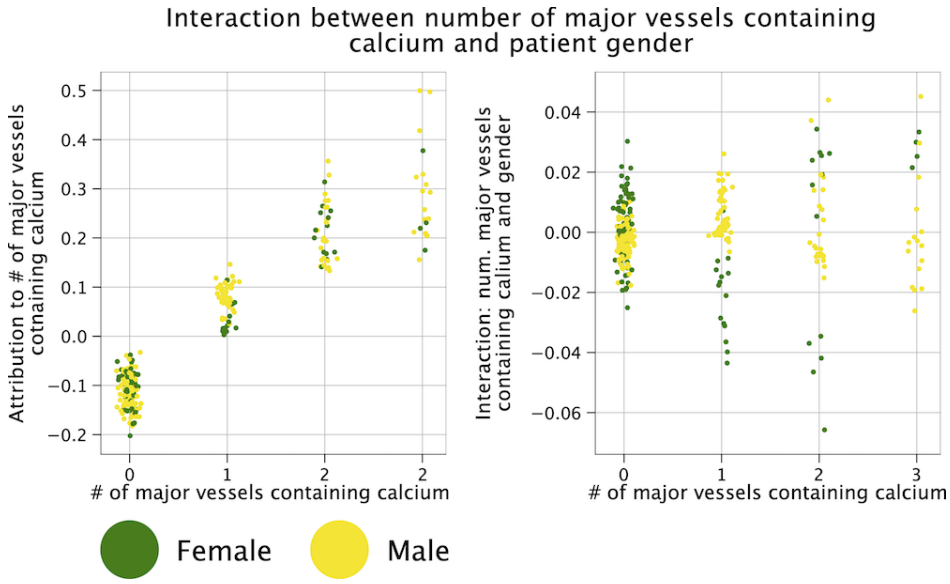

Figure 3.23: Left: Expected Gradients feature importance of the number of major vessels with accumulation of calcium as indicated by cardiac cinefluoroscopy. More vessels with calcium build-up indicated increased risk. Right: Expected Hessians feature interactions between patient gender and the number of major vessels containing calcium. When the Expected Hessians interactions are aggregated across the data set, they reveal that our model has learned that women with calcium deposition in one coronary artery are less likely than men to be diagnosed with coronary artery disease.

interference. The DM-SNR curve measures the degree to which phase correction changes the signal-to-noise ratio in the measured signal. Since pulsars are far away, their radio emissions get dispersed as they travel from the star to earth: low frequencies get dispersed more than high frequencies (e.g., they arrive later). Phase correction attempts to re-sync the frequencies; however, no amount of phase correction should help peak a signal if the signal was generated from radio frequency interference rather than a legitimate pulsar.

On this task, we used a two-layer neural network with 32 hidden units in both layers and the SoftPlus activation function after each layer. We optimized using stochastic gradient descent with a batch size of 256. We used an initial learning rate of 0.1, which decays with a rate of 0.96 every batch, and nesterov momentum with $\beta = 0.9$ [276]. We trained for 10 epochs and used a class-weight ratio of 1:3 negative to positive to combat the imbalance in the training data set. Again, we note that these hyper-parameters are not necessarily optimal but were simply chosen because they produced reasonable convergence on the training set. We split the data into 14,318 training examples (1,365 are pulsars) and 3,580 testing examples (274 are pulsars) and achieved a held out test accuracy of 0.98 (0.86 TPR and 0.99 TNR).

To generate attributions and interactions for this data set, we used Expected Gradients and Expected Hessians, with the training set forming the background distribution. We used 200 samples to compute both attributions and interactions, although 200 samples was probably larger than necessary. In Figure 3.24, we examine the interaction between two key features in the data set: kurtosis of the integrated profile, which we abbreviate as kurtosis (IP), and standard deviation of the dispersion-measure signal-to-noise ratio curve, which we abbreviate as standard deviation (DM-SNR). The bottom of Figure 3.24 shows that kurtosis (IP) is a highly predictive feature, while standard deviation (DM-SNR) is less predictive. However, in the range where kurtosis (IP) is roughly between 0 and 2, standard deviation (DM-SNR) helps distinguish between a concentration of negative samples at standard deviation (DM-SNR) $<$ 40. We can verify that the model we trained correctly learns this interaction. By plotting the interaction values learned by the model against the value of kurtosis (IP), we see a peak positive interaction for points in the indicated range and with high standard deviation (DM-SNR). Interaction values show us that the model has successfully learned the expected pattern: that standard deviation (DM-SNR) has the highest discriminative power when kurtosis (IP) is in the indicated range.

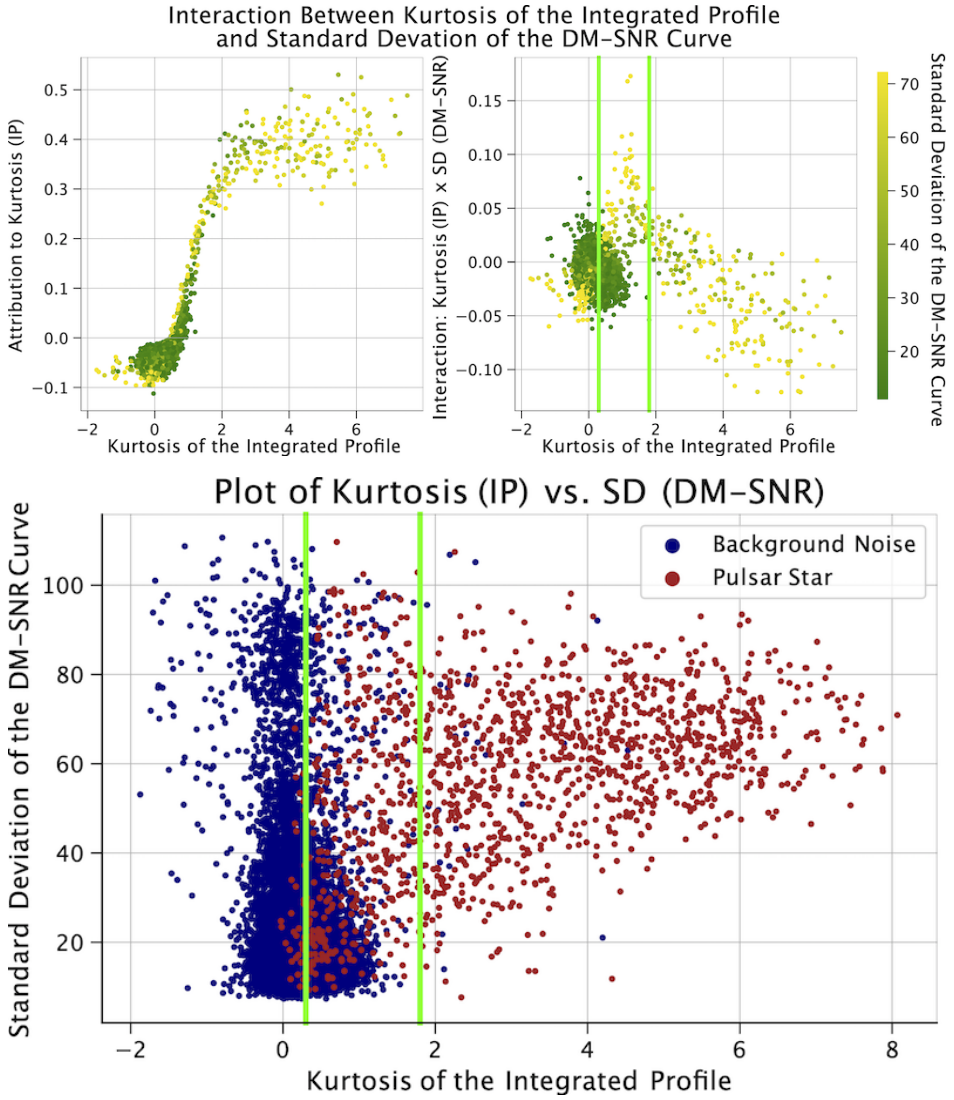

Figure 3.24: Top Left: Attributions to kurtosis (IP) generated by expected gradients. Top Right: The model learns a peak positive interaction when kurtosis (IP) is in the range [0, 2]. Bottom: A plot of the training data along the axes of the two aforementioned features, colored by class label. Although kurtosis (IP) seems to be the more predictive feature, in the highlighted band the standard deviation (DM-SNR) provides useful additional information: a larger standard deviation (DM-SNR) implies a higher likelihood of being a pulsar star.

3.13 Supplement: Details for Anti-Cancer Drug Combination Response Prediction

3.13.1 Data Description

As mentioned in Section 6.2, our data set consisted of 12,362 samples (available from the CTD2 data portal). Each sample contained the measured response of a 2-drug pair tested on the cancer cells of a patient [286]. The 2-drug combination was described by both a *drug identity indicator* and a *drug target indicator*. For each sample, the drug identity indicator was a vector $x_{\text{id}} \in \mathbb{R}^{46}$ where each element represented one of the 46 anti-cancer drugs present in the data; each element took a value of 0 if the corresponding drug was not present in the combination and a value of 1 if the corresponding drug was present in the combination. Therefore, for each sample, x_{id} had 44 elements equal to 0 and 2 elements equal to 1, the most compact possible representation for the 2-drug combinations. The drug target indicator was a vector $x_{\text{target}} \in \mathbb{R}^{112}$, where each element represented one of the 112 unique molecular targets of the anti-cancer drugs in the data set. Each entry in this vector equalled 0 if neither drug targeted the given molecule, equalled 1 if one of the drugs in the combination targeted the given molecule, and equalled 2 if both drugs targeted the molecule. The targets were compiled using the information available on DrugBank [296]. The *ex vivo* samples of each patient's cancer was described using gene expression levels for each gene in the transcriptome, as measured by RNA-seq, $x_{\text{RNA}} \in \mathbb{R}^{15377}$. Before training, the data was split into two parts – 80% of the samples were used for model training, and an additional 20% were used as a held-out validation set to determine when the model had been trained for a sufficient number of epochs.

3.13.2 RNA-seq Preprocessing

The cancerous cells in each sample were described using RNA-seq data, i.e., measurements of the expression level of each gene in the sample. We describe here the preprocessing steps used to remove batch effects while preserving biological signals. We first converted raw transcript counts to fragments per kilobase of exon model per million mapped reads (FPKM), a measure known to better reflect the molar amount of each transcript in the original sample than raw counts. FPKM accounts for this by normalizing the counts for different genes according to the length of transcript, as well as for the total number of reads included in the sample [193]. The equation for FPKM is

$$\text{FPKM} = \frac{X_i \times 10^9}{N l_i}, \tag{3.56}$$

where X_i is the vector containing the number of raw counts for a particular transcript i across all samples, l_i is the effective length of that transcript, and N is the total number of

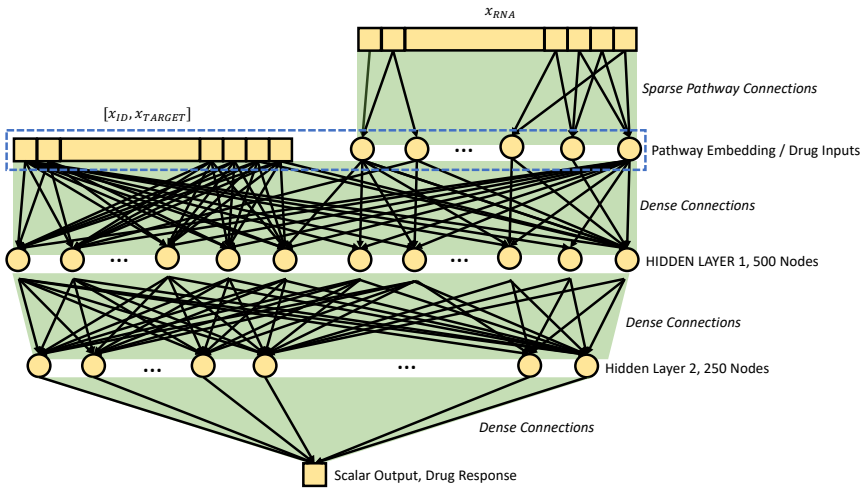

Figure 3.25: Neural network architecture for anti-cancer drug combination response prediction. We learn an embedding from all RNA-seq gene expression features (x_{RNA}) to KEGG pathways by sparsely connecting the inputs only to nodes corresponding to the pathways of which they are members. When we calculate feature attributions and interactions, we attribute to the layer that contains the raw drug inputs and the learned pathway embeddings (layer boxed with dashed blue line).

counts. After converting raw counts to FPKM, we opted to consider only the protein-coding part of the transcriptome by removing all non-protein-coding transcripts from the data set. Protein-coding transcripts were determined according to the list provided by the HUGO Gene Nomenclature Committee). In addition to non-protein-coding transcripts, we also removed any transcript that was not observed in $> 70\%$ of the samples. Transcripts were then $\log_2$ transformed and made 0-mean, unit variance. Finally, the ComBat tool (a robust empirical Bayes regression implemented as part of the sva R package) was used to correct for batch effects [159].

3.13.3 Model and Training Description

To model the data, we combined the successful approaches of [220] and [97]. Our network architecture was a simple feed-forward network (Fig. 3.25), as in [220], where there were two hidden layers of 500 and 250 nodes, respectively, both with Tanh activation. To improve performance and interpretability, we followed [97] in learning a *pathway-level* embedding of the gene expression data. The RNA-seq data, $x_{\text{RNA}} \in \mathbb{R}^{15377}$, was sparsely connected

to a layer of 1077 nodes, where each node corresponded to a single pathway from KEGG, BioCarta, or Reactome [119, 202, 47]. We made this embedding non-linear by following the sparse connections with a Tanh activation function. The non-linear pathway embeddings were then concatenated to the drug identity indicators and the drug target indicators, and these served as inputs to the densely connected layers.We trained the network to optimize a mean squared error loss function and used the Adam optimizer in PyTorch with default hyperparameters and a learning rate equal to 10^{-5} [131]. We stopped the training when mean squared error on the held-out validation set failed to improve over 10 epochs, and found that the network reached an optimum at 200 epochs. For easier calculation and more intuitive attribution, we attributed the model's output to the layer with the pathway embedding and drug inputs rather than to the raw RNA-seq features and drug inputs (see Fig. 3.25).

For this experiment, we calculated all explanations and interactions using the Integrated Gradients and Integrated Hessians approach, using the all zeros vector as reference and $k > 256$ interpolation points.

3.13.4 Biological Interaction Calculation

To evaluate how well the interactions detected by Integrated Hessians matched with the ground truth for biological drug-drug interactions in this data set, we can use additional single drug response data (that our model was not given access to) in order to calculate *biological synergy*. Drug synergy is the degree of extra-additive or sub-additive response observed when two drugs are combined as compared to the additive response that would be expected if there were no interaction between the two compounds. The drug response for a single drug is measured as $\text{IC50}^{\text{single}}$, or the dose of that single drug necessary to kill half of the cells in an *ex vivo* sample. The drug response for a drug combination is measured as $\text{IC50}^{\text{combination}}$, or the dose of an equimolar combination of two drugs necessary to kill half of the cells in an *ex vivo* sample. The drug synergy between two drugs a and b can be calculated using the CI, or combination index:

$$CI_{a,b} = \frac{\text{IC50}_a^{\text{combination}}}{\text{IC50}_a^{\text{single}}} + \frac{\text{IC50}_b^{\text{combination}}}{\text{IC50}_b^{\text{single}}}. \tag{3.57}$$

For CI, a value greater than 1 indicates anti-synergy (negative interaction), while a value less than 1 indicates synergy (positive interaction). While our model was trained solely to predict $\text{IC50}^{\text{combination}}$, we determined how well the model learned true biological interactions by using the additional single drug response data to calculate synergy for particular samples. As described in Section 6.2, when we calculated CI values for all samples in which the combination of Venetoclax and Artemisinin was tested, then binarized the samples into synergistic and anti-synergistic, we saw that the Integrated Hessians values were higher in

the truly synergistic group than in the truly anti-synergistic group ($p = 2.31 \times 10^{-4}$). This is remarkable given that the model had no access to single drug data whatsoever.

Chapter 4

AI FOR RADIOGRAPHIC COVID-19 DETECTION SELECTS SHORTCUTS OVER SIGNAL

4.1 Introduction

The prospect of applying artificial neural networks to the detection of COVID-19 in chest radiographs has generated interest from machine learning (ML) researchers and radiologists alike, given its potential to (i) help guide management in resource-limited settings that lack sufficient numbers of the gold-standard reverse-transcription polymerase chain reaction (RT-PCR) assay, and (ii) clarify cases of suspected false negatives from the RT-PCR assay [194, 146]. While numerous recent publications and preprints report machine learning models with high performance at this task [84, 291, 100, 209, 30, 121], the trustworthiness of these models needs to be rigorously evaluated before deployment in a clinical setting [150].

Our findings in this study support the troubling possibility that these models fail to learn the true underlying pathology reflecting the presence of COVID-19 and instead leverage spurious associations between presence or absence of COVID-19 and radiographic features that reflect variations in image acquisition, *i.e.*, "shortcuts" [79]. While such spurious associations may arise in any dataset, we observed that many recent ML models for radiographic detection of COVID-19 were trained using data with the potential for near *worst-case* confounding: these datasets are composed of an exclusively COVID-19 negative source and a COVID-19 positive source, such that any systematic differences between the sources correlate perfectly with COVID-19 status [84, 291, 100, 209, 30, 121]. Similar combinations of data sources, where the source label correlates with disease status, have also been used to train AI systems for detection of COVID-19 in computed tomography scans [98] (though the non-public nature of the data precludes experimental verification of the extent of shortcut learning in this setting) and for other medical imaging tasks [152, 4], implying that our findings have broad implications to the field of medical machine learning.

In this study, we evaluate the trustworthiness of recent deep learning models for COVID-19 detection from chest radiographs. After training deep convolutional neural networks [104, 142] (Methods, Fig. 4.7) in the manner of these previous publications [84, 291, 100, 209, 30, 121], we evaluate their performance in new hospital systems. Then, we interrogate the extent to which these models rely on confounds by identifying the most important image features using state-of-the-art explainable AI techniques, including both saliency maps and generative

adversarial networks (GANs) [274, 313, 261, 65]. These inquiries reveal how seemingly high-performance AI systems may derive the majority of their performance from the exploitation of undesired shortcuts, highlighting the need to verify that AI systems rely on the desired signals. Finally, we evaluate several methods to alleviate the problem of shortcut learning in this setting, demonstrating the importance of improved data quality for the creation of robust and useful models.

4.2 Results

4.2.1 Overview of the experimental approach

Before examining our main results, we first outline our experimental approach (Fig. 4.1a). To begin, we reviewed the literature to examine the datasets and models used for detection of COVID-19 from chest radiographs, with attention toward studies with the potential for "worst-case confounding." After choosing representative networks, we build two datasets: one that reproduces the data used in previous studies, and a second that enables external validation on new hospitals. In a first experiment, we evaluate models that were trained on one dataset using test images from the other dataset, under the expectation that a model that relies on valid medical pathology–which should not change between datasets–should maintain high performance. We then probe deeper into specific shortcuts that these models leverage, using techniques from explainable AI.

In a "model-centric" approach, which focuses on the specific portions of the radiographs that contribute most to the predictions of our models in particular, we build saliency maps using Expected Gradients [65]. In essence, this approach attributes importance to each pixel of a radiograph based on the gradients of our models, while avoiding issues such as saturation or an arbitrary choice of baseline. We complement this model-centric approach with a data-centric approach, which focuses on the key aspects of the data that could be used to distinguish COVID-19 positive and COVID-19 negative cases. Specifically, we apply generative adversarial networks (CycleGANs[313]) to transform COVID-19 positive radiographs to appear COVID-19 negative and vice versa, in the sense that key image features are transformed, such that a network can no longer discriminate between the real images of a given pathology label and the transformed images from the opposite class[261]. Rather than use our classifier networks to perform this discrimination task, we instead train new discriminator networks simultaneously with generator networks that transform the images, such that this experiment focuses on key aspects of our data, rather than our classifiers in particular.

To further validate these findings, we go on to perform "region-swapping" experiments, in which we swap out portions of radiographs that our explainable AI approaches identify as

important, with the expectation that changes to truly important regions will have a large impact on our classifiers' outputs. We conclude by evaluating approaches to mitigate shortcut learning from the perspectives of both generalization performance and model explainability.

4.2.2 Literature review of model and dataset construction

In our investigation, we aimed to determine the extent to which shortcut learning affects AI systems for COVID-19 detection in chest radiographs, which is complicated by the diversity of these systems. We therefore trained a series of ten models with varied architectures, including state-of-the-art networks that were tailor-made for detection of COVID-19 in chest radiographs[310, 291, 209] and multiple "off-the-shelf," general-purpose architectures.[104, 241, 301, 142] For our primary models, we chose a network based on the DenseNet-121 architecture[104], which we judged faithfully replicated the modeling choices of recent high-performance models for COVID-19 classification, while also following established best practices for classification of pathologies from chest radiographs using deep learning. Alongside these primary models, we also investigate multiple secondary models, to help probe the generality of our findings and the extent to which they apply to AI systems found in the wild. These secondary models include: the COVID-Net network, which was custom designed for detection of COVID-19 via a machine-based architecture search[291]: the DarkCovidNet model, which was modified from a standard Darknet-19 model for the purpose of COVID-19 detection[209]; and the CV19-Net model[310], which was built by ensembling twenty DenseNet-121 networks and motivates our primary model, which uses the same architecture without ensembling, given that ensembling did not provide performance gains but substantially increases computational complexity (see Results Section 4.2.3 "Evaluation of models on new hospital systems").

To train and evaluate these models, we created two datasets (Fig. 4.1a). Dataset I consisted of COVID-19 positive radiographs from the GitHub-COVID repository [43], which aggregates radiographs from publication figures and other online sources with varied geographic origin. We supplemented these with COVID-19 negative radiographs from the National Institutes of Health's (NIH) ChestX-ray14 repository [292], which originates from a single hospital in the United States.

Dataset I is similar to the datasets used for training in recent publications on AI for COVID-19 detection [291, 100, 84, 209, 121, 30]. Specifically, four of these publications[84, 100, 209, 30] combine the GitHub-COVID repository with either the NIH repository[292] or the similar Radiological Society of North America pneumonia dataset [205], which was derived from the NIH repository; two others[291, 121] similarly combine these repositories and then supplement with additional COVID-19+ images from other online repositories, many of which have since been added to the GitHub-COVID repository. Given the continually evolving nature of many of these repositories, the precise set of images used in each study

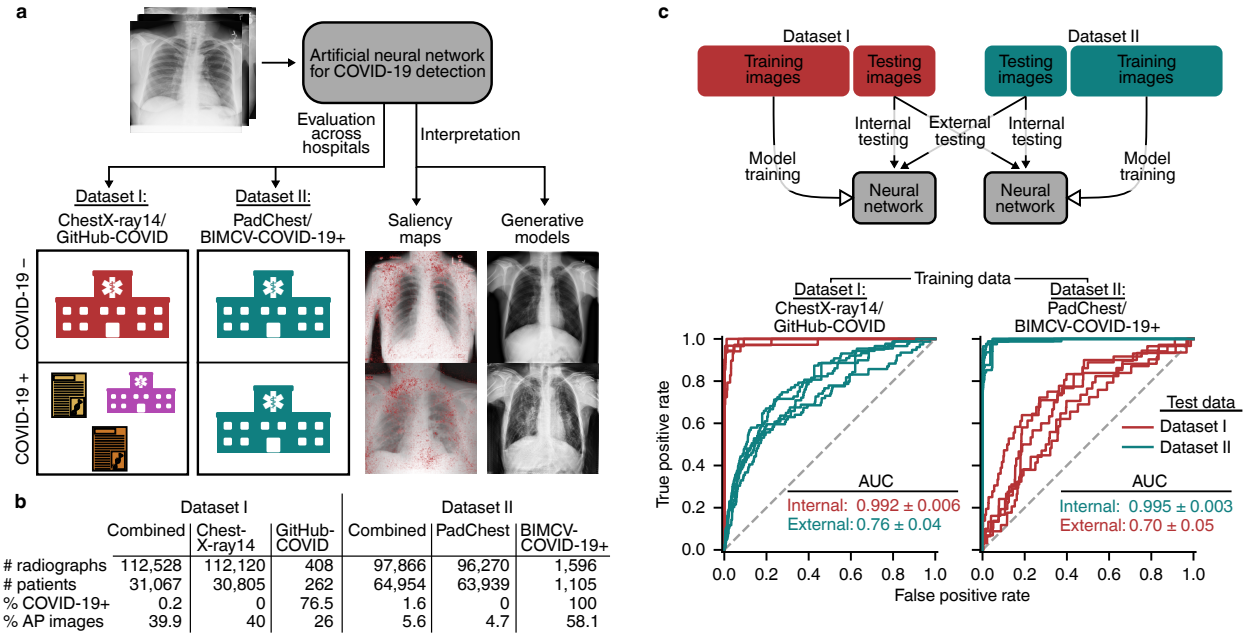

Figure 4.1: **Overview of the study design.** **a**, A neural network model is trained to detect COVID-19 using radiographs from either of two datasets, and then evaluated on both datasets to learn how performance may drop in deployment (i.e., a generalization gap). Intepretability methods are then applied to infer what the model learned and which features were important for its decisions. Whereas Dataset I draws radiographs from multiple hospital systems as well as cropped images from publication figures, Dataset II draws radiographs from multiple hospitals from a single regional hospital system. **b**, Characteristics of the datasets used in this study. **c**, Model evaluation scheme (top) and corresponding receiver operating characteristic (ROC) curves (bottom), which indicate the performance of our neural network models evaluated on both an *internal* test set (new, held-out examples from the same data source as the training radiographs) and an *external* test set (radiographs from a new hospital system). Inset numbers indicate area under the ROC curves, where larger area corresponds to higher performance (AUC, mean ± standard deviation). The difference between internal and external test set performance is the generalization gap.

remains unclear, and additional uncertainty is introduced by the dearth of documentation on the source of some images or the validity of their labels (*e.g.*, in the ActualMed and Figure 1 databases at `https://github.com/agchung/Actualmed-COVID-chestxray-dataset` and `https://github.com/agchung/Figure1-COVID-chestxray-539dataset`). This uncertainty notwithstanding, our core observation is that numerous well-cited studies build their datasets by gathering COVID-19+ radiographs from varied sources, as exemplified most thoroughly by the GitHub-COVID repository (in which the image sources and labeling method are clearly documented), and then combining these with COVID-19 negative radiographs originating from the NIH repository, such that we judge our Dataset I fairly represents the key aspects of the data used in these prior works. Other publications[310, 294, 160, 196], which generally use non-public data that precludes our ability to audit their models, do not share this issue of strong correlation between data source labels and COVID-19 status, but based on our review of the literature, we find this issue in an alarming proportion of the publications, including many of the most high profile studies[291, 100, 209].

Unlike the datasets used in recent publications, which collected COVID-19 positive and negative images from disparate sources, Dataset II corresponds to a seemingly more ideal case where both COVID-19 positive and negative images were drawn from similar sources. This dataset, which comprises the PadChest and BIMCV-COVID-19+ repositories (Fig. 4.1a-b), consisted of radiographs from a single region and published by a shared research team, though BIMCV-COVID-19+ represents a greater diversity of hospitals than PadChest, and the repositories were acquired over different time periods [33, 288].

4.2.3 Evaluation of models on new hospital systems

After training on Dataset I, we evaluated our models for reliance on confounding factors by comparing the predictive performance on an internal test set (new, held-out radiographs from Dataset I) to performance on external radiographs from Dataset II. While our models attain high performance on internal test data, *half of the model's predictive performance is lost* when testing on Dataset II (Fig. 4.1c, left). This performance drop (i.e., generalization gap) suggests these models rely on source-specific confounds in the radiographs, as we would expect models that use genuine markers of pathology to generalize well [79]. This finding held true for all nine additional architectures we examined, including those that were custom tailored in recent studies for detection of COVID-19 in radiographs (Fig. 4.8-4.9).

While we initially expected that a dataset built from radiographs drawn from a single region would be less likely to contain spurious correlations that enable ML models to take shortcuts, we found that models trained on Dataset II also exhibit high performance on internal test data and low performance on external test data (Fig. 4.1c, right, and Fig. 4.8). Thus, dataset-level confounding may pose a severe issue even in datasets derived from

more similar sources, such as hospitals from a single region, contrary to the conclusions of contemporary work [184]. These findings argue for routine reporting of metadata on potential patient, hospital system, and preprocessing confounds. By illuminating the construction of radiographic datasets in greater detail, these data will make it easier for domain experts to identify likely sources of confounding. Additionally, these metadata enable the construction of models that explicitly control for confounds, providing a route to AI systems that generalize well even in the context of confounded training data [36, 113]. In contrast, we note that a popular set of approaches to improve generalization performance, known as "unsupervised domain adaptation," are precluded by the presence of worst-case confounding because these methods rely on learning models invariant to data-source labels, which will be perfectly correlated with the pathology labels [77].

4.2.4 Alternate hypotheses do not explain poor generalization

To verify the hypothesis that exploitation of dataset-specific confounding leads to poor generalization performance, we investigated alternative explanations for the generalization gap. Previous publications have suggested that more complex models, *i.e.*, those with higher *capacity*, may be particularly prone to learning confounds [240], so we evaluated the generalization performance of simpler models, including a logistic regression and a simple convolutional neural network architecture, but found that the generalization gap did not improve (Fig. 4.9). This result further supports the broad applicability of our findings, since the generalization gap was present regardless of network architecture, aligning with a previous study which showed that radiograph classification performance is robust to neural network architecture [28]. Likewise, we found that replacing the multi-label classification scheme of our original models with a simpler single-label classification scheme (see Methods Section 4.1) did not improve generalization performance.

In addition to the choice of model architecture, an alternative explanation for poor generalization performance is that, rather than the model learning a spurious correlation that does not generalize, the model learns a genuine relationship between a radiograph's appearance and its COVID-19 label that still does not generalize. One such scenario is that the COVID-19 detection task differs between training and test-time, which may occur in our datasets given that most of the images in the GitHub-COVID dataset were cropped from scientific publications and thus are perhaps more likely to show radiographic evidence of COVID-19, while labels in the BIMCV dataset are derived solely from RT-PCR or serology, and therefore may or may not feature radiographic evidence of COVID-19. However, when we modified the label scheme of BIMCV-COVID-19+ such that radiographs are only labelled positive if a radiologist noted evidence of COVID-19, the generalization gap persisted (Fig. 4.10), suggesting that such *concept shift* between training and test time does not explain the

performance difference and leaving the use of spurious correlations as the best explanation [223].

4.2.5 Explainable AI identifies spurious confounders

We further interrogated the trained AI models using saliency maps [225, 192, 274], which highlight the regions of each radiograph that contribute most to the model's prediction (Supplementary Note and Fig. 4.11), to determine specific confounds that deep convolutional networks for COVID-19 detection exploit. While our saliency maps sometimes highlight the lung fields as important (Fig. 4.2a), which suggests that our model may take into account genuine COVID-19 pathology, the saliency maps concerningly also highlight regions outside the lung fields that may represent confounds. The saliency maps frequently highlight laterality markers that originate during the radiograph acquisition process (Fig. 4.2a and Fig. 4.12), which differ in style between the COVID-19-negative and COVID-19-positive datasets, and similarly highlight arrows and other annotations that are uniquely found in the publication-sourced radiographs of the GitHub-COVID data source [43] (Fig. 4.13), which aligns with a previous study finding that ML models can learn to detect pneumonia based on spurious differences in text on radiographs [305]. Our saliency maps also indicate that the image edges, the diaphragm, and the cardiac silhouette are important for our models' predictions of a patient's COVID-19 status, though these regions are *not* among those routinely used by radiologists to assess for COVID-19 [200] and instead likely reflect dataset-level differences in patient positioning and radiographic projection, *i.e.*, anterior-posterior (AP) vs. posterior-anterior (PA) view [113]. Reliance on such confounds, which do not consistently correlate with COVID-19 status in outside datasets, helps explain the previously observed poor generalization performance.

To further investigate what features could be used by an ML model to differentiate between the COVID-19 positive and COVID-19 negative datasets, we trained generative adversarial networks (GANs) to transform COVID-19 negative radiographs to resemble COVID-19 positive radiographs and vice versa. This technique should capture a broader range of features than saliency maps, as the GANs are optimized to identify all possible features that differentiate the datasets. Consistent with our knowledge of how radiologists detect evidence of COVID-19 in chest radiographs, the GAN increases the radiopacity or radiolucency of the lung fields bilaterally to respectively add or remove evidence of COVID-19, indicating that neural network models are capable of learning genuine markers of COVID-19 (Fig. 4.2b, blue boxes, and Figs. 4.14 and 4.15). However, the generative networks frequently add or remove laterality markers and annotations (Fig. 4.2b, solid red boxes), reinforcing our observation from saliency maps that these spurious confounds also enable ML models to differentiate the COVID-19 positive and COVID-19 negative radiographs. The

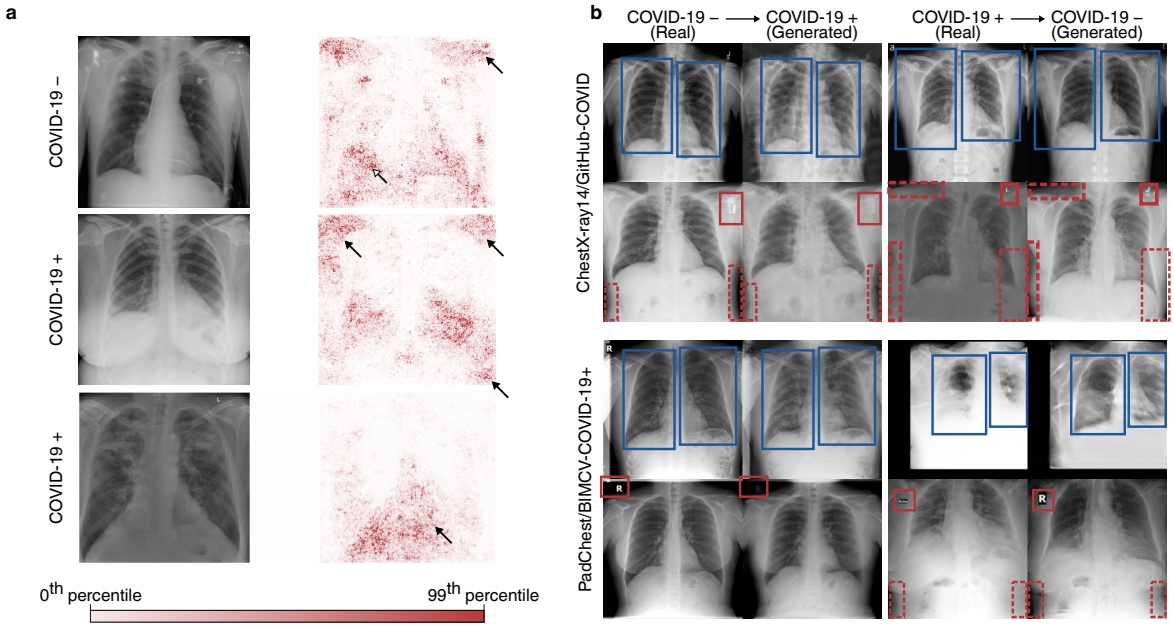

Figure 4.2: **Explainable AI visualizes image factors important for deep neural networks trained to detect COVID-19 in radiographs. a**, Saliency maps for our neural network models indicating the regions of each radiograph with the greatest influence on the model's prediction. Top, in a COVID-19 negative radiograph, in addition to the highlighting in the lung fields (open arrow), the saliency maps also emphasize laterality tokens (closed arrow). Middle, in a COVID-19 positive radiograph, the most intensely highlighted regions of the image are the bottom corners (arrows) outside of the lung fields. Bottom, in a COVID-19 positive radiograph, the only highlighted region is the diaphragm (arrow). Colorbar indicates saliency map pixel importances by percentile. (Caption continued on next page. **b**, Radiographs and their corresponding transformations by a generative adversarial network (GAN), illustrating systematic differences that enable neural networks to differentiate between COVID-19 positive and negative radiographs. COVID-19 negative images are transformed by the GAN to appear as if they were COVID-19 positive, and vice versa. Comparison of images before and after transformation with a GAN visualizes important image features for COVID-19 prediction. Blue boxes indicate alterations to the opacity of the lung fields, which may represent the network's attention to genuine COVID-19 pathology. Red solid boxes indicate altered laterality markers, and red dashed boxes indicate altered radiopacity at the image borders, both of which may spuriously correlate with a patient's COVID-19 status in the training data. Image credit for bottom image of panel **a** (not modified) and bottom right images of upper half of panel **b** (modified by annotation and transformation by CycleGAN): Winther, H., Laser, H., Gerbel, S., Maschke, S., Hinrichs, J., Vogel-Claussen, J., Wacker, F., Hoeper, M., & Meyer, B.; CC-BY 3.0 license, doi:10.6084/m9.figshare.12275009.

generative networks additionally alter the radiopacity of image borders (Fig. 4.2b, dashed red boxes), supporting our previous assertion that systematic, dataset-level differences in patient positioning and radiographic projection provide an undesirable shortcut for ML models to detect COVID-19. Given this strong evidence that ML models can leverage spurious confounds to detect COVID-19, we also investigated the extent to which our classifiers, in particular, relied upon the features altered by the GAN. We found that images transformed by the GANs were reliably predicted by the classifiers to be the transformed class rather than the original class (Fig. 4.16), demonstrating that the majority of features used by our classifiers were altered by the GAN, *i.e.*, the features identified by the GAN are approximately a superset of those used by the classifiers. Thus, the image transformations from the GANs enable us to see hypothetical versions of the same radiographs that would have caused our classifiers to predict the opposite COVID-19 status.

4.2.6 Experimental validation of factors identified by interpretability methods

We next aimed to experimentally validate the importance of spurious confounds to our models by manually modifying key features (Fig. 4.3a-b). We first swapped laterality markers from a COVID-19 positive and COVID-19 negative image, and found that introduction of a laterality marker more common in COVID-19 positive images increased the models' predicted odds that the patient had COVID-19, while the converse also held. As a control, we compared to randomly swapped image patches of the same size and found that the change in model output from swapping laterality markers is significantly greater than expected by random (Fig. 4.3a), indicating that laterality markers are key features leveraged by our models to determine a patient's COVID-19 status. While these markers vary consistently between the datasets (Fig. 4.4, 4.13, 4.14, and 4.15), these markers would not reliably indicate COVID-19 status in more general settings. We similarly investigated the shoulder region of radiographs, which was frequently highlighted as an important feature in our saliency maps (Fig. 4.13), and found that moving the clavicle region of a radiograph to the top border of the radiograph increased the model's predicted odds that the patient has COVID-19 (Fig. 4.3b and 4.17), suggesting that the models leverage the consistent but medically irrelevant difference in patient positioning between the COVID-19 negative and COVID-19 positive data sources. To verify whether these findings held on a population basis, we sampled a random subset of the radiographs and repeated our experiments involving the swapping of laterality markers and movement of the shoulder region (Fig. 4.18), which confirmed that our models indeed leverage these shortcuts throughout the dataset.

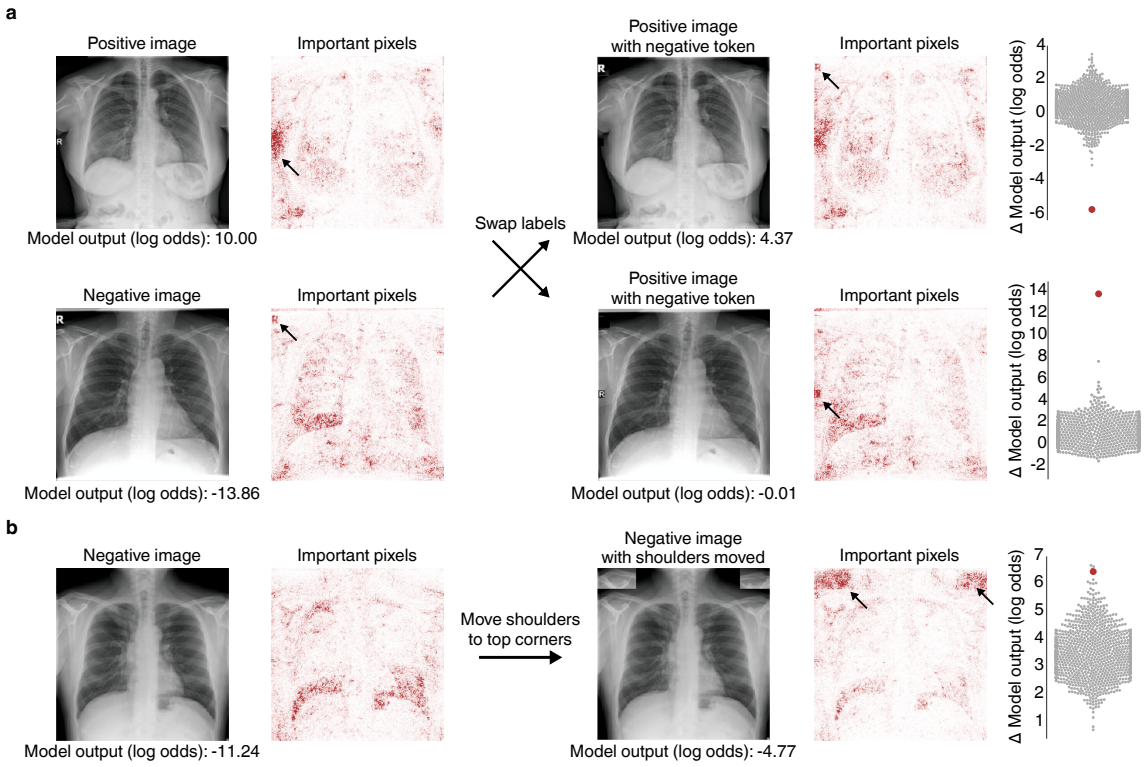

Figure 4.3: **Experimental confirmation of insights from saliency maps and CycleGANs via radiograph modification. a**, (Left) Text markers on radiographs are highlighted by saliency maps as important for COVID-19 prediction. The exchange of laterality markers between a pair of COVID-19 + and COVID-19 - images significantly shifts the output when compared to swapping random patches of the same size: Δ positive image (log odds) $= -5.63$ (empirical p-value $= 9.99 \times 10^{-4}$ based on Monte Carlo substitution of random image patches, n=1000); Δ negative image (log odds) $= 13.85$ ($p = 5.00 \times 10^{-3}$, n=1000) (Methods Sections 4.5 and 4.6). Gray dots in the distribution plots (right) correspond to the change in model output after swapping random image patches, which were used as a negative control, while the red dots correspond to the change in model output for the radiographs with swapped laterality markers. **b**, Positioning of patient shoulders may impact COVID-19 prediction. Saliency maps highlight the shoulder region as important predictors of COVID-19 positivity after (but not before) this region is moved to the top of the image (left). This patch increased model output significantly more than random patches of the same size moved to the same corners ($\Delta = 6.57$, empirical p-value $= 5.00 \times 10^{-3}$, n=1000). Gray dots in the distribution plot (Right) correspond to radiographs with randomly selected patches, while the red dot corresponds to the radiograph with the shoulder regions moved.

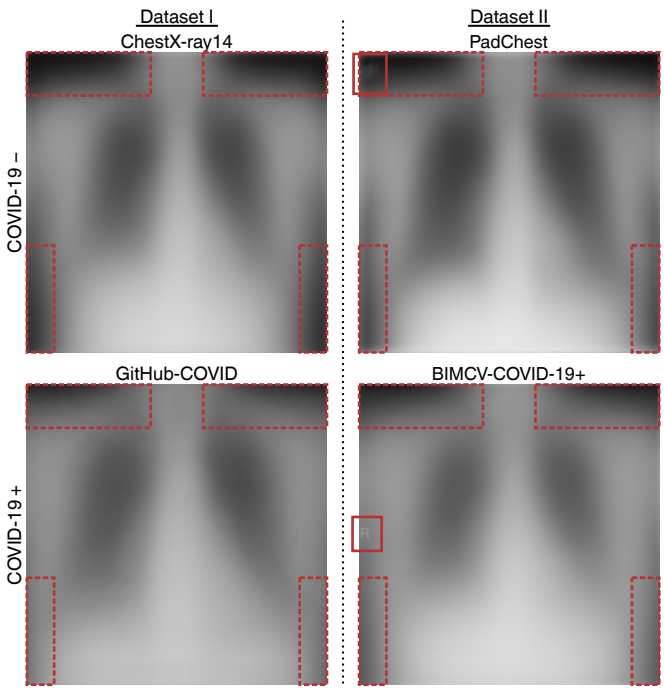

Figure 4.4: **Average images from the four repositories used to construct datasets in this study, demonstrating systematic differences between the radiograph repositories that could be exploited by AI systems.** Solid red boxes indicate systematic differences in laterality markers that are visible in the average images, and dashed red boxes indicate systematic differences in radiopacity of the image borders, which could arise from variations in patient position, radiographic projection, or image processing.

4.2.7 Shortcuts have a variable effect on generalization

Importantly, some shortcuts will impair generalization performance, while other shortcuts will not; while the large generalization gap is explained well by shortcut learning, a portion of the remaining external test set performance may still be due to shortcuts that happen to generalize for our datasets. Both types of shortcut are undesirable, since even those that generalize between our datasets may not consistently generalize to other settings, and the use of clinical rather than strictly radiological information extracted from these radiographs may be redundant, depending on the clinical workflow.

To analyze which shortcuts may contribute to poor generalization, we considered clinical metadata and average images from each repository (Fig. 4.4). Among the shortcuts that do not generalize are the textual markers, which were clearly identified by our explainability approaches as important for prediction of COVID-19 but appear differently in the COVID-19 negative and COVID-19 positive images from each repository (Fig. 4.4). In addition, the radiographic projection, which may contribute to (but does not completely explain) the importance of the image edges and shoulder position, does not generalize between the datasets (Fig. 4.1b, "% AP images" row) and therefore may contribute to poor generalization performance.

Among the shortcuts that do generalize (at least between our datasets) are aspects of patient positioning that do not result from the radiographic projection. These aspects of patient positioning also likely contribute to the previously observed importance of image edges and shoulder position, and they maintain a consistent relationship with COVID-19 negative and COVID-19 positive radiographs in each dataset (Fig. 4.4), despite the inconsistent relationship of the radiographic projection with COVID-19 status. An additional factor that may generalize well is patient sex, since within both datasets, a higher proportion of males were COVID-19 positive. Taken together with our observation that half of our models' performance is attributable to confounds that do not generalize well, we conclude that only a minority of our models' performance is attributable to monitoring for genuine COVID-19 pathology.

Given that radiographic projection and patient sex are diffusely represented in radiographs and therefore less clearly pointed out by our explainability approaches, we additionally validated whether our models could leverage these factors as shortcuts. We reasoned that for a model to be able to leverage these concepts as shortcuts, the same model (when retrained) must be able to predict these concepts well. Indeed, our models accurately predict both the radiographic projection and patient sex for both internal and external test data (Fig. 4.5), which supports that these concepts are easily learned and available to be leveraged as shortcuts. Considering that these concepts are easily learned and are also predictive of COVID-19 status (*i.e.*, they are correlated with COVID-19 in our datasets), we judge that

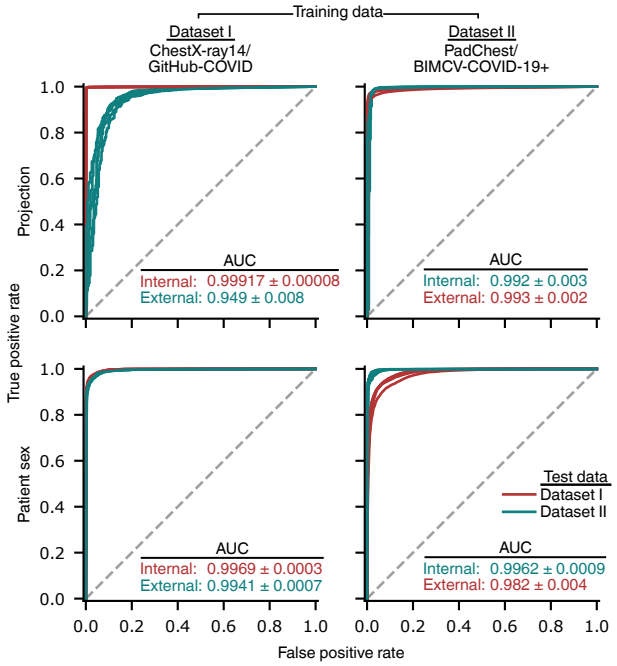

Figure 4.5: **Evaluation of the extent to which the prediction of image factors that could be leveraged as shortcuts to detection of COVID-19 generalizes to new hospitals.** Models were trained to predict radiographic projection (AP vs. PA view) and then evaluated on internal and external test radiographs. Inset values indicate area under the ROC curve (AUC, mean ± standard deviation, n=5).

our networks likely incorporate this information to predict COVID-19 status.

4.2.8 Improved data mitigates shortcut learning

Given this strong evidence that neural networks leverage dataset-level differences as shortcuts for COVID-19 status, we inquired to what extent this issue might be mitigated. While an initial hypothesis may be that the choice of neural network architecture determines the propensity for shortcut learning, all architectures that we examined displayed similar evidence for shortcut learning, as quantified by the generalization performance (Fig. 4.8). While our tests hinted that data augmentation may help alleviate shortcut learning, the effect was small and not statistically significant (Fig. 4.8b; external test set ROC-AUC of 0.76 $\pm$ 0.04 vs. 0.79 $\pm$ 0.03 before and after data augmentation, respectively, when trained on dataset I, p = 0.22, $U = 6$ based on a Mann-Whitney U-test; external test set ROC-AUC of 0.70 $\pm$ 0.05 vs 0.69 $\pm$ 0.05 before and after data augmentation, respectively, when trained on dataset II, p = 1.00, $U = 13$ using Mann-Whitney U-test).

In principle, an attractive solution to mitigate shortcut learning is to remove the image factors that the models leverage as shortcuts. However, in practice, it is difficult to remove all such image factors. As a simple test case, we inquired whether removing textual markers by cropping to the center 75% of each radiograph would reduce shortcut learning and thus improve generalization performance. After retraining our models on these cropped radiographs, we found that such cropping does not improve generalization performance (Fig. 4.19), which naïvely may suggest that these textual markers do not contribute to shortcut learning. However, considering the consistent identification of this factor by saliency maps, the CycleGANs, and manual image modifications (Fig. 4.2a-b, Fig. 4.3a), a more likely explanation is that a multitude of redundant shortcuts exist, such that a model may shift its attention toward other shortcuts in absence of a particular shortcut; conjecturally, such image attributes could include the size of the lung fields relative to the image, the positioning of the scapular shadows, the size of the cardiac silhouette, image intensities, or textural features that enable inference of the data source.

Perhaps a more reliable solution to remove the image factors that enable shortcut learning is to simply collect data that is less confounded. To test this hypothesis, we created a third dataset (Dataset III) to represent a nearly optimal case; the COVID-19 positive and negative cases were taken from the BIMCV-COVID-19+ repository and its paired BIMCV-COVID-19− repository (`https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/`), respectively, which were collected from the same hospitals over the same time period (Fig. 4.20). If this near-optimal dataset solved the "shortcut problem," then we would expect that models trained on these data may (i) attain higher performance on an external test set, since bonafide pathology should transfer between datasets while shortcuts may or may not, and (ii) exhibit

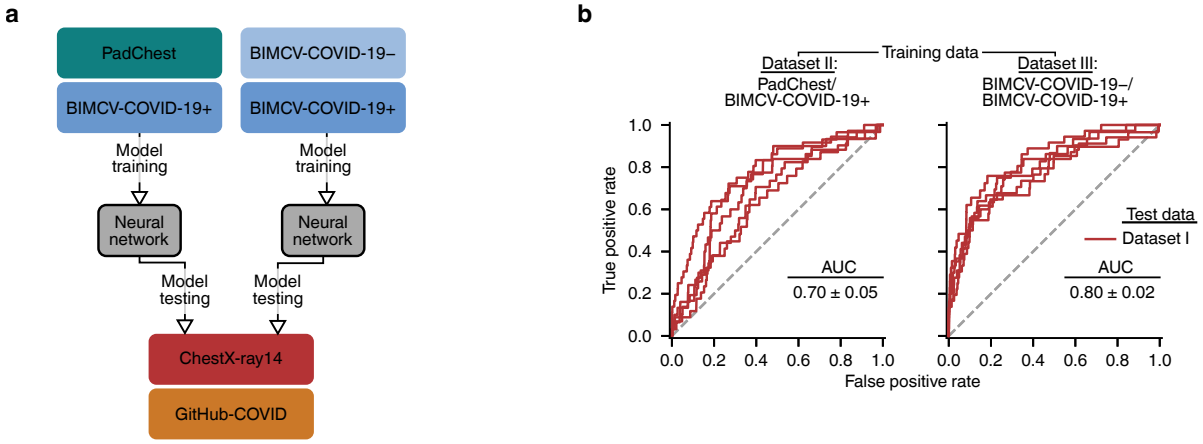

Figure 4.6: **Mitigation of shortcut learning via collection of improved data. a**, To evaluate whether improved data collection mitigates shortcut learning, we train classifiers on dataset II and dataset III, then test both on the same external data (dataset I). **b**, Evaluation of generalization performance as measured by receiver operating characteristic (ROC) curves. Inset values indicate area under the ROC curve (AUC, mean ± standard deviation, n=5). $^{*}p = 0.016$ based on a two-tailed Mann-Whitney U-test (corresponding U=-2.4).

a lower generalization gap, in the sense that performance on an internal test set would not as drastically misrepresent the true performance, as measured on external data. We trained models to detect COVID-19 in Dataset III and then tested these models on external data from Dataset I, and compared these results to models that were trained on Dataset II and tested on Dataset I. Despite that Dataset III contains approximately $1/20^{th}$ the images of Dataset II, it attains significantly higher performance on external data (Fig. 4.6), and exhibits little generalization gap (Fig. 4.21), suggesting that collection of less confounded data indeed alleviates the issue of shortcut learning. Furthermore, saliency maps for the model trained on Dataset III tend to attribute more importance to the lung fields, where COVID-19 pathology would be expected, than to potentially confounding regions, as compared to the equivalent saliency maps generated for the model trained on Dataset II (Fig. 4.22), though the saliency maps still show some attention toward shortcuts. Taken together, these findings argue for careful collection of data so as to minimize potential for shortcut learning, with continued caution that improved data collection may only partially solve the problem.

4.3 Discussion

ML models that were built and trained in the manner of recent studies generalize poorly and owe the majority of their performance to the learning of shortcuts. This undesired behavior owes partially to the synthesis of training data from separate datasets of COVID-19 negative and COVID-19 positive images, which introduces near worst-case confounding and thus abundant opportunity for models to learn these shortcuts. Importantly, since undesirable "shortcuts" may be consistently detected in both internal and external domains, our results warn that external test set validation alone may be insufficient to detect poorly behaved models.

Previous studies also audited AI systems for detection of COVID-19 in radiographs, with mixed success at identification of shortcuts. In a simple yet clever approach, one study found that models retain high performance when examining only the borders of radiographs, such that genuine COVID-19 pathology was removed from the images[184]. This study concurs with our findings but comments primarily on the possibility of this issue rather than its occurrence in the wild, though it is nonetheless alarming. The study that introduces the COVID-Net model also audits its model, using a saliency map approach known as "GSInquire," but in contrast does not identify evidence of shortcut learning in a set of three published images[291]. Given the similarity of that study's training data to our own Dataset I and the large generalization gap that we observe with the same architecture, we suspect shortcut learning likely indeed occurred, and it remains unclear whether auditing decisions about additional radiographs beyond the three presented would have revealed evidence of shortcut learning, or if the GSInquire approach, which is not available through a public-facing repository, fails to identify the shortcuts. A number of other studies that involve datasets with severe confounding between pathology and image source [84, 100, 209, 30, 121] similarly audit their models using saliency map approaches (most prominently, the Grad-CAM approach [250]) and report findings on one to three radiographs, without noting evidence of shortcut learning. Based on this pattern, we recommend that researchers examine and report results from explainable AI or saliency map approaches on a population level, employing a sampling-based approach as necessary, and to remain skeptical of high performances in the absence of external validation. Moreover, we find that population-level audits using saliency maps are highly labor intensive to perform in a rigorous manner and may depend on domain knowledge, which motivates future approaches for explainable AI in medical imaging that simplify population-level analysis.

Our findings support common-sense solutions to alleviate shortcut learning in AI systems for radiographic COVID-19 detection, including (i) improved collection of training data, *i.e.*, data in which radiographs are collected *and processed* in a way matching the target population of a future AI system and (ii) improved choice of the prediction task to involve

more clinically relevant labels, such as a numeric quantification of the radiographic evidence for COVID-19 [160, 298]. However, we demonstrate that shortcut learning may occur even in a more ideal data collection scenario, highlighting the importance of explainable AI and principled external validation. While AI promises eventual benefits to radiologists and their patients, our findings demonstrate the need for continued caution in the development and adoption of these algorithms [150].

4.4 Methods

4.4.1 Model architecture and training procedure

For our primary neural network, we used a convolutional neural network with the DenseNet-121 architecture to predict the presence versus absence of COVID-19 [104]. This architecture has not only been used in a variety of recent models for COVID-19 classification [291, 100], but has also been used for the diagnosis of non-COVID pneumonia [225, 113], as well as for more general radiographic classification [76].

Following the approach in recent COVID-19 models [291, 100], we first pre-trained the model on ImageNet, a large database of natural images [53]. Forcing models to first learn general image features should also serve as an inductive bias to prevent overfitting on domain-specific features [113]. After ImageNet pre-training, the final 1000-node classification layer of the trained ImageNet model was removed and replaced by a 15-node layer, corresponding to the 14 pathologies recorded in the ChestX-ray14 dataset plus an additional node corresponding to COVID-19 pathology; while only the prediction for COVID-19 was used for evaluating the model, we followed previous works that showed simultaneous learning of multiple tasks was useful for achieving highest predictive performance [225]. To obtain a consistent label scheme, labels in the GitHub-COVID, PadChest, and BIMCV-COVID-19+ repositories were mapped to the 14 ChestX-ray14 categories.

The model was optimized end-to-end using mini-batch stochastic gradient descent with a batch size of 16, momentum parameter of 0.9, weight decay of 10^{-4}, and learning rate of 0.01, which was decreased by a factor of 10 every 5 epochs. We chose a binary cross entropy loss as the optimization criterion. To prevent overfitting, we monitored the area under the ROC curve (AUROC) for COVID-19 classification on a held-out validation set, and chose the epoch with the highest validation AUROC as the final model. All models were trained for 30 epochs, which was long enough for all models to reach a maximum in the validation AUROC. All models were trained using the PyTorch software library [211], version 1.4, on NVIDIA RTX 2080 TI graphics processing units and required approximately 5 hours of training time per replicate.

We additionally examined three architectures that were designed in previous publications

specifically for the task of COVID-19 detection, with the hypothesis that these specialized architectures may better learn genuine COVID-19 pathology and generalize better to external data. These architectures include CV19-Net[310], DarkCovidNet[209], and COVID-Net[291]. We trained these models on datasets I and II, following the image pre-processing procedures, data augmentation pipelines, and optimization schemes used in the original publications (we note that while dataset I is analogous to the original datasets used to train DarkCovidNet and COVID-Net, CV19-Net was trained on data that is not publicly available). For both CV19-Net and DarkCovidNet, the base architectures were dowloaded from the torchvision library[210], then modified to match the descriptions in each respective paper. The COVID-Net network was adapted from an open-source, PyTorch implementation (by Ilias Papastratis; `https://github.com/iliasprc/COVIDNet`). For the CV19-Net paper, the data augmentation pipeline was altered to match the pipeline in the original paper: when loading images, each radiograph is additionally randomly flipped with probability 0.5 then rotated between -30 and 30 degrees. To disentangle performance differences due to the ensembling present in the CV19-Net architecture from performances differences due to the change in data augmentation, we also trained a single DenseNet-121 model with the same data augmentation steps as the CV19-Net. In the case of the CV19-Net and DarkCovidNet, we maintained the same multilabel classification task (*i.e.*, the 14 ChestX-ray14 labels plus a label for COVID-19) to facilitate optimal comparison between architectures. In the case of the COVID-Net architecture, due to problems with vanishing and exploding gradients when using the full multilabel classification task, we reduced our full label set to only the 3 labels used in the COVID-Net paper (COVID-19 Pneumonia, Non-COVID Pneumonia, No Pneumonia). We also trained additional, popular architectures that were not tailored specifically for COVID-19 detection, including MobileNetv2[241] and ResNeXt-50[301]. These networks were again modified from the ImageNet-pretrained base models in the torchvision library [210]. We trained these architectures using the same pre-processing scheme and optimization parameters as our DenseNet-121 models, again replacing the standard, 1000-label classification layers with an analogous layer for our 15 labels.

To test the hypothesis that lower-capacity models may not learn spurious correlations [240], we also trained two lower-capacity models. The first, an AlexNet model [142], was trained in the same manner as the DenseNet-121, with the weights randomly initialized rather than pretrained on ImageNet. The second was a logistic regression with "deep features": since individual pixels do not have stable semantic meaning over different samples in the dataset, we first extract a set of 1024 higher-level features using the feature embedding (*i.e.*, the activations of the penultimate layer) of a DenseNet-121 trained on ImageNet and then fit a logistic regression to these fixed features. This procedure is accomplished by training the DenseNet-121 architecture with the weights of its feature embedding subnetwork frozen. The AlexNet and logistic regression were optimized using the same training parameters as the full

DenseNet-121 model specified above. The fact that lower-capacity models did not generalize better in our setting may be due to the fact that Sagawa et al. focus on a reweighted training scheme [240], while our models were trained to minimize empirical risk in order to replicate the training schemes used by recent COVID-19 detection models (see above).

4.4.2 Datasets and preprocessing

To train and evaluate our models, we combined images from five large open-access repositories of chest radiographs into three datasets (Fig. 4.1a). The first, which we refer to as Dataset I, was designed to replicate the datasets used to develop and evaluate the most popular COVID-19 diagnostic models [291]. In this dataset, we collected COVID-19 negative images from the NIH ChestX-ray14 repository, representing 112,120 radiographs from 30,805 patients from the NIH Clinical Center [292]. We collected COVID-19 positive images from the GitHub-COVID repository [43] (commit ID 9b9c2d5), representing 408 radiographs from 262 patients, where this data was originally collected from figures in scientific publications and assorted web sources of COVID-19 positive cases.

The second dataset, which we refer to as Dataset II, was designed to represent a more ideal case in terms of domain confounding – both COVID-19 positive and COVID-19 negative images were acquired from hospitals from a common region and were published by a shared research team. We collected COVID-19 negative images from the PadChest repository, representing 96,270 radiographs from 63,939 patients from a hospital in Valencia, Spain [33]. The COVID-19 positive images in our dataset were taken from the BIMCV-COVID-19+ dataset (version 1), which represents 1,596 images from 1,015 patients (after exclusions), from the same regional hospital system in Valencia, Spain [288]. We note that while PadChest and BIMCV-COVID-19+ originate from the same region, potential for confounding remains since (i) PadChest was collected from a single hospital whereas BIMCV-COVID-19+ was collected from multiple hospitals, and (ii) the repositories were collected over different time periods, over which image acquisition techniques may have changed.

The third dataset, referred to as Dataset III, was designed to represent the most ideal case in terms of domain confounding. Unlike dataset II, the COVID-19 positive and COVID-19 negative images were collected from not only the same region, but from the same hospitals and over the same time period. Like dataset II, the COVID-19 positive images were collected from the BIMCV-COVID-19+ repository. The COVID-19 negative images were taken from the corresponding BIMCV-COVID-19− repository, which includes 3086 images from 2327 patients (after exclusions).

Following the recommendations by Cohen et al. [43], we filtered radiographs from the online repositories to include only PA and upright AP radiographs. Lateral radiographs, AP supine radiographs, radiographs with unknown projections, and computed tomography scans

were excluded from the datasets. Images with absent radiographic windowing information, which was necessary to display radiographs from the BIMCV-COVID-19+ and BIMCV-COVID-19− repositories, were also excluded.

We partitioned each repository into training, validation, and test folds, ensuring that all radiographs of any given patient belong to a single fold. Since the ChestX-ray14 dataset specifies a "test" partition, we used these radiographs as part of our dataset I test fold. Of the remaining portion, 5% were reserved as a validation fold, while the rest were used directly for training. In the PadChest, BIMCV-COVID-19+, and BIMCV-COVID-19− repositories, we reserved 5% of the radiographs for testing, and 5% of the remaining radiographs for validation. Due to the smaller size of the GitHub-COVID repository, we reserved 10% of the radiographs for testing, and 10% of the remaining radiographs for validation. With the exception of the ChestX-ray14 test fold, which was held fixed as explained above, the folds were drawn at random for each model replicate.

4.4.3 Model interpretability using saliency maps

To generate saliency maps, which enable interpretation of machine learning models by assigning importance values to each pixel of an input image, we apply a state-of-the-art approach known as *Expected Gradients* [65]. Broadly, this approach captures the notion of "importance" by tracking how each pixel of an image impacts the output of the model when contrasted with a set of noninformative baseline examples, where the impact is measured by accumulating the model's gradients (a mathematical measure of a model's sensitivity to small changes in a feature) as the image is interpolated from the baseline example to the image of interest. Formally, the Expected Gradients attribution ϕ for an input sample x and input feature i is defined:

$$\phi_i(x) := \mathop{\mathbb{E}}_{x' \sim D, \alpha \sim U(0,1)} \left[(x_i - x'_i) \times \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \right], \tag{4.1}$$

where D represents a *background distribution* from which reference samples x' are drawn. This method is an extension of the popular saliency map approach Integrated Gradients, which is the special case of Expected Gradients in which there is only a single reference sample.

For our application, Expected Gradients improves over Integrated Gradients in terms of the accuracy of its saliency maps [65] and the inclusion of multiple reference samples, which avoids the choice of a single reference that may be arbitrary but nonetheless impactful upon the resultant saliency maps [268]. Finally, path-based approaches like Expected Gradients and Integrated Gradients are preferable to other methods for generating saliency maps because they are theoretically principled: these methods are provably guaranteed to attribute

importance to important pixels and guaranteed not to attribute importance to unimportant pixels (also see Supplementary Note) [274].

As the background distribution D for Expected Gradients, we used the COVID-19-negative images from the training dataset for each model we explain. Intuitively, we are explaining how the output of our model for our input image x differs on average from the output of the model for images in the training data D. We demonstrate that Expected Gradients is not overly sensitive to choice of D by comparing the saliency maps for several radiographs with a background distribution of images from the training data to attributions for those same radiographs with a background distribution of images from the external dataset, and found the resultant attributions are similar (Fig. 4.23).

4.4.4 Data interpretability using CycleGAN

To attain visual explanations of the differences between COVID-19 positive and COVID-19 negative images in each dataset, we aimed to understand which characteristics of the chest radiograph would have to change to make a COVID-19 negative image appear to be a COVID-19 positive image, and vice versa. Formally, let $\mathcal{X}$ be a domain of COVID-19 negative images, and let $\mathcal{Y}$ be a domain of COVID-19 positive images. Our goal is to learn a mapping $G : \mathcal{X} \mapsto \mathcal{Y}$ that takes a COVID-19 negative chest radiograph, $X \in \mathcal{X}$, and transforms it so that it is indistinguishable from COVID-19 positive chest radiographs. We also aim to learn the inverse transformation, $F : \mathcal{Y} \mapsto \mathcal{X}$.

Since generative adversarial networks have previously been shown to be effective for the interpretation of neural networks, we learn these two transformations using the CycleGAN approach [261, 313]. The mappings G and F are learned by two neural networks, which are optimized in conjunction with two discriminator networks $D_{\mathcal{Y}}$ and $D_{\mathcal{X}}$. These networks are optimized to minimize a series of losses. The first, referred to as the *adversarial loss*, encourages the mapping functions G and F to match the distribution of generated images from each source domain to the true data distribution of each target domain:

$$\mathcal{L}_{\text{GAN}}(G, D_{\mathcal{Y}}, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{Y \sim p_{\text{data}}(Y)}[\log D_{\mathcal{Y}}(Y)] + \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log(1 - D_{\mathcal{Y}}(G(X))], \tag{4.2}$$

$$\mathcal{L}_{\text{GAN}}(F, D_{\mathcal{X}}, \mathcal{Y}, \mathcal{X}) = \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log D_{\mathcal{X}}(X)] + \mathbb{E}_{Y \sim p_{\text{data}}(Y)}[\log(1 - D_{\mathcal{X}}(F(Y))], \tag{4.3}$$

where $p_{\text{data}}(X)$ and $p_{\text{data}}(Y)$ represent the data distributions for each domain. In addition to the adversarial loss, the networks are also trained to enforce *cycle consistency*, meaning that $F(G(X)) = X$. This is desirable, since it enforces a similarity between the original and transformed images. The loss here is:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{X \sim p_{\text{data}}(X)}[\|F(G(X)) - X\|_1] + \mathbb{E}_{Y \sim p_{\text{data}}(Y)}[\|G(F(Y)) - Y\|_1]. \tag{4.4}$$

The full loss that is optimized then is simply the sum of these three losses:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}}(G, D_{\mathcal{Y}}, \mathcal{X}, \mathcal{Y}) + \mathcal{L}_{\text{GAN}}(F, D_{\mathcal{X}}, \mathcal{Y}, \mathcal{X}) + \mathcal{L}_{\text{cyc}}(G, F) \tag{4.5}$$

To understand which image features are important in distinguishing the domains $\mathcal{X}$ and $\mathcal{Y}$, we transform a COVID-19 negative radiograph $X \in \mathcal{X}$ or a COVID-19 positive radiograph $Y \in \mathcal{Y}$ using the learned generator networks G or F to map the image to the opposite domain. We then compare which image features are changed in the transformation.

Our CycleGAN networks were implemented in Python 3.7 using the PyTorch software library and an open-source implementation of the CycleGAN approach (by Aitor Ruano; `https://github.com/aitorzip/PyTorch-CycleGAN`). To attain comparable training time, the networks for trained for 3000 epochs (Dataset I) or 1000 epochs (Dataset II). Each network required approximately one week of training time on an NVIDIA RTX 2080 graphics processing unit.

4.4.5 Experimental validation of feature attributions

We experimentally validated our findings from saliency maps and GANs by modifying important radiographic features. To detect whether the higher-level features that our saliency maps highlight are major contributors to the model's classification, we used methods inspired by a behavioral testing approach [231]. For example, saliency maps highlight dataset-specific laterality markers and text within the images. If these text markers are indeed important, then moving a marker from a COVID-19 positive image to a COVID-19 negative image should increase the predicted log odds of COVID-19. For a pair of COVID-19 positive and COVID-19 negative images, we swap the text markers and measure the change in the output for each image. To assess the significance of the change in the model's output at the level of each individual image, we generate empirical p-values by comparing to a null distribution generated by swapping 1,000 random patches of each image of the same dimensions as the text markers (Fig. 4.3a). We conduct a similar experiment to validate whether the shoulder regions frequently highlighted in the saliency maps have a significant impact on the model's decisions. We observe that the shoulder region of COVID-19 positive images tends to appear at the upper image border, while the shoulder region of COVID-19 negative images appears slightly lower. Furthermore, the saliency maps highlight the clavicles and shoulders of the COVID-19 positive images, but not in the COVID-19 negative images. We hypothesized that the model was looking for the presence of shoulders in the upper corners of the image. To test our hypothesis, we moved the clavicles and shoulders of a COVID-19 negative image to the top corners of the radiograph and measured the change in model output (Fig. 4.3b). We tested for statistical significance at the level of individual images by generating empirical p-values. Our distribution was generated by randomly sampling and replacing 1000 patches of

the same size as the shoulder region, following the same procedure described for the laterality markers.

In order to verify the significance of these regions for our models at a *population level*, we repeated the procedure described in the paragraph above for a sample of randomly selected radiographs from the datasets (see Fig. 4.18). For the dataset-specific laterality markers (Fig. 4.18, left), we randomly sampled 10 COVID-19 negative images with laterality or other text markers and 10 COVID-19 positive images with laterality or other text markers. To test for the significance of the text markers across the datasets, we used a Wilcoxon signed rank test to compare the distribution of the magnitudes of changes in model output after swapping the text markers to the distribution of the magnitudes of the average changes in model output after swapping 1000 random patches of the same size ($p = 8.86 \times 10^{-5}$, Siegel's T statistic $= 0.0$). For the positioning of the shoulder regions (Fig. 4.18, right), we randomly sampled 20 COVID-19 negative images. We then used a Wilcoxon signed rank test to compare the distribution of changes in model output after moving the clavicles and shoulder regions to the top of the image with the distribution of the average changes in model output after moving 1000 random patches of the same size ($p = 8.86 \times 10^{-5}$, Siegel's T statistic $= 0.0$).

4.4.6 Statistics

In our experiments involving manual modification of radiographs (Fig. 4.3a-b, 4.17), we computed empirical p-values by first generating the distribution of the change in the model output (in log odds space) for a set of random, non-specific modifications as described in each caption. The p-value was then calculated as $(r + 1)/(n + 1)$ where r is the number of non-specific modifications that produced a greater increase in model output (greater magnitude decrease in Fig. 4.3a, top row) and n is the total number of non-specific modifications [203].

To compare the generalization performance of models (*e.g.*, Fig. 4.6), we performed a two-tailed Mann-Whitney U-test, given that the ROC-AUC values are bounded by 0 and 1 and therefore unlikely to be normally distributed.

4.5 Data availability

All radiographs are compiled from publicly-available data repositories. The ChestX-ray14 repository is available at `https://nihcc.app.box.com/v/ChestXray-NIHCC`. The GitHub-COVID dataset is available at `https://github.com/ieee8023/covid-chestxray-dataset`. The PadChest repository is available at `https://bimcv.cipf.es/bimcv-projects/padchest/`. The BIMCV-COVID19 repositories are available at `https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/`.

4.6 Code availability

All of the code necessary to reproduce our experimental findings can be found at `https://github.com/suinleelab/cxr\_covid` (archived at `https://doi.org/10.5281/zenodo.4623792`).

4.7 Acknowledgments

This work was funded by the National Science Foundation [CAREER DBI-1552309 to S.-I.L.] and the National Institutes of Health [R35 GM 128638 and R01 AG061132 to S.-I.L.]. We thank Hugh Chen and Gabriel Erion for providing feedback while writing the manuscript. We thank Dr. Aurelia Bustos for clarifying characteristics of the PadChest and BIMCV-COVID-19+ datasets. We also thank Dr. David Janizek for insight into the interpretation of COVID-19 on chest radiographs.

Supplementary Information

Supplementary Note

While saliency maps are widely used to interpret image-based artificial intelligence systems [225, 192, 20], the reliability of these approaches has been disputed by contemporary work, which observes that saliency maps explaining medical imaging classifiers fail to localize medically relevant pathology [7]. However, this prior work did not disentangle whether (i) the saliency maps fail to identify the features that are important for the classification models, or (ii) the saliency maps faithfully identify the features that are important for the classification models, but the models do not depend on medically relevant pathology. We hypothesised the latter, that attribution maps fail to localize relevant pathology because the models they explain do not rely on relevant pathology [83].

To validate that the pixels selected by our saliency maps are truly important for the models they explain, we chose 100 images that our model predicted are COVID-19 negative, then masked and mean-imputed a subset of pixels. If we selected these pixels at random, we would expect the models output to regress to the mean output (become more positive) since the negative images become more like the mean image (which is predicted to be more positive than the COVID-19 negative images). If the pixels identified by Expected Gradients are important for the model's prediction, we would anticipate that masking these pixels should make the model's output *more positive* than masking randomly selected pixels. When we mask the top 10% of pixels identified by EG as contributing to the negative prediction of the model, we see that the model's output is shifted to be significantly more negative than when we mask pixels selected at random (Supplementary Fig. 4.11).

Supplementary Figures

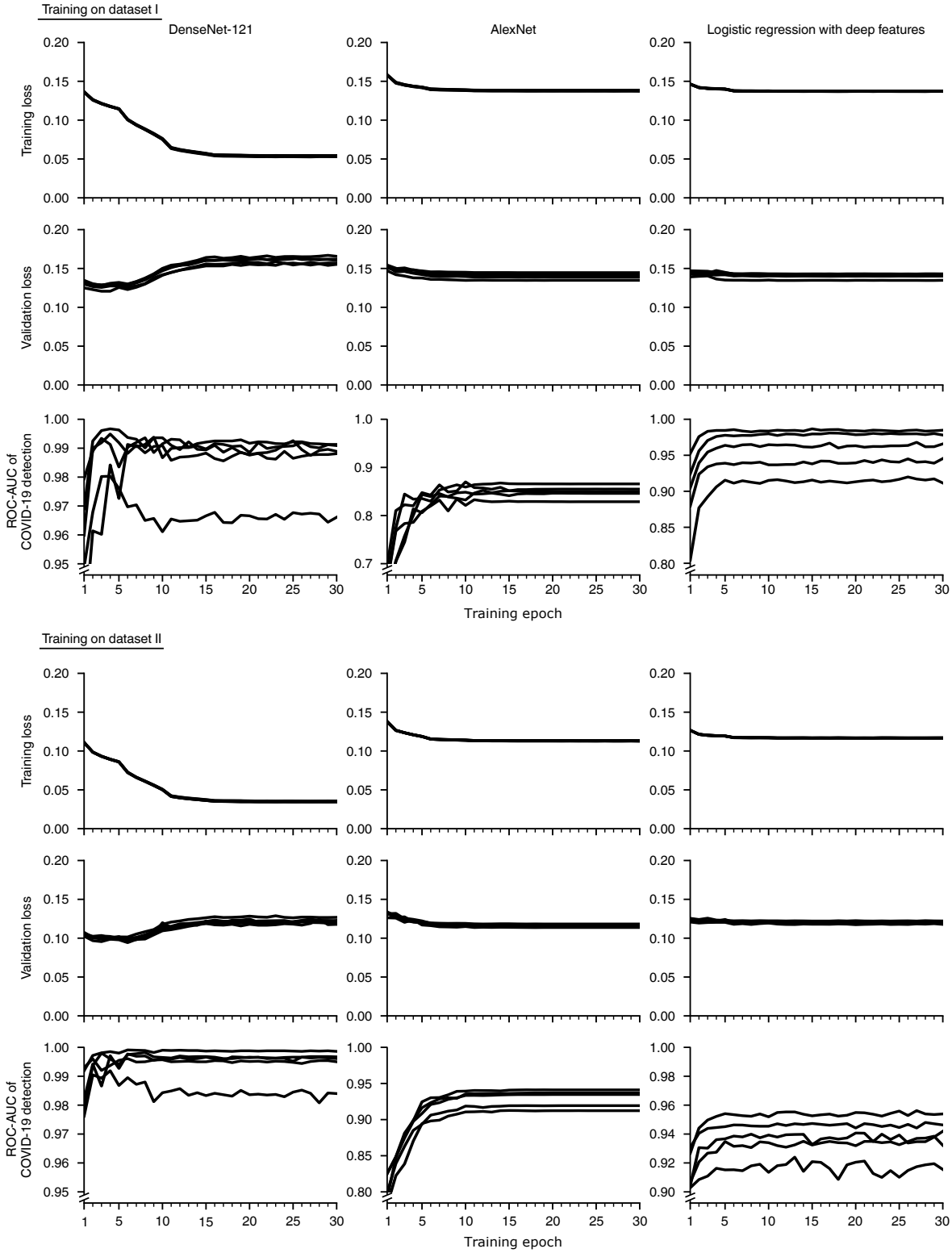

Figure 4.7: (Caption next page.)

Figure 4.7: **(Previous page.) Evolution of metrics that monitor the artificial neural network training process.** Training curves are shown for each of 5 random train/validation/test splits of the datasets. During the training procedure, the model is progressively optimized to decrease the training loss, for which we chose the *binary cross entropy*. The validation loss monitors the same metric on a subset of the training radiographs that is held-out from the optimization process (and that is also entirely separate from testing data). Increases in the validation loss may indicate that the model has *overfit* the training data, *i.e.*, the model has memorized the training data rather than learning general principles that apply to new radiographs, such as those in the validation set. To prevent overfitting, we save models when they achieve a maximum in the area under the receiver operating characteristic curve (ROC-AUC) for COVID-19 classification in the held-out validation set, and we use these models for all subsequent analysis. All models were trained for a total of 30 epochs, which was sufficient to attain a maximum in the ROC-AUC of COVID-19 classification. Note that to permit visualization of the maximum in the ROC-AUC of COVID-19 detection, the plots that visualize this quantity feature variable y-axis scales.

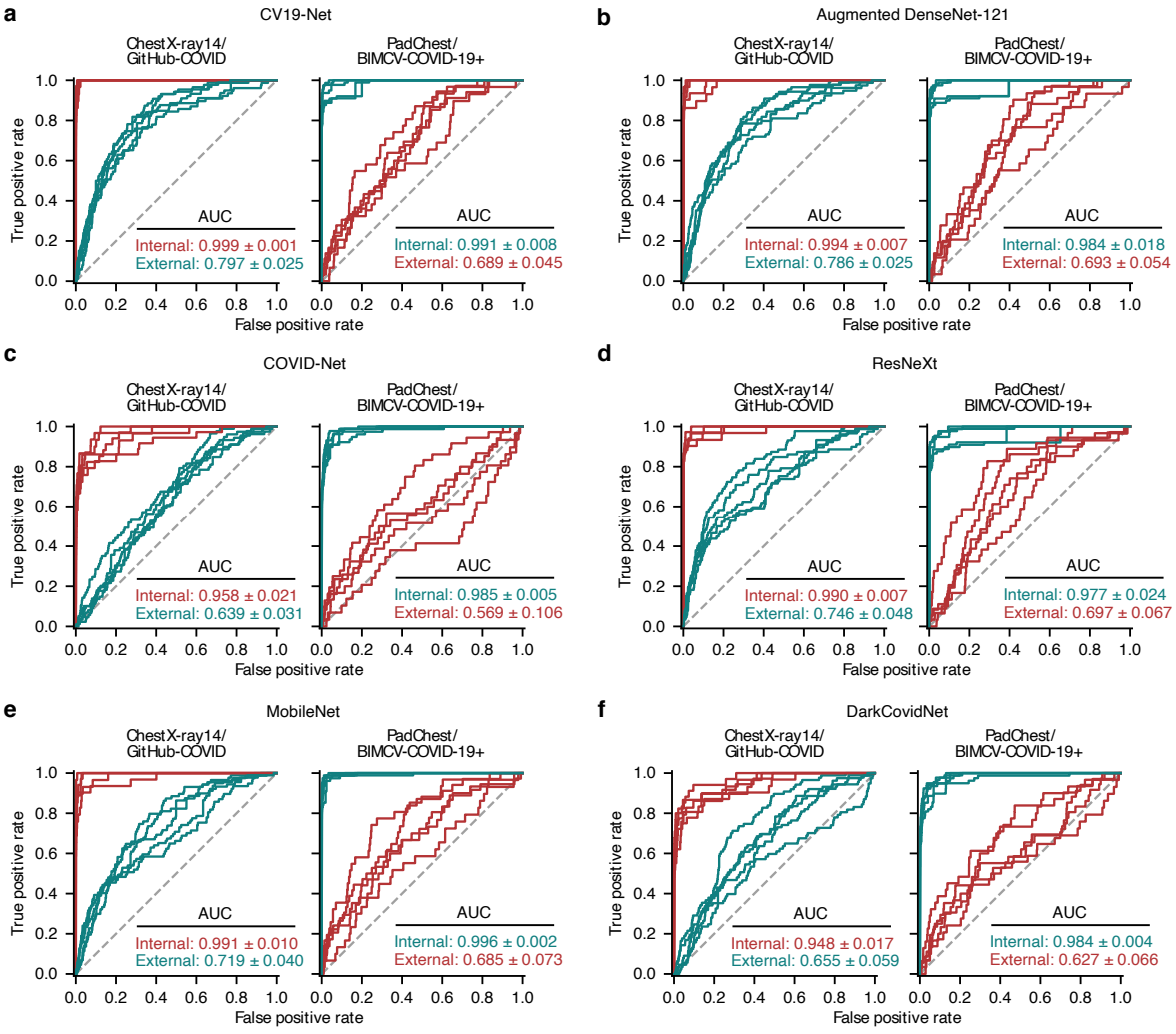

Figure 4.8: **Generalization performance of models that were specifically designed in previous studies for detection of COVID-19 in chest radiographs as well as additional "off-the-shelf" architectures.** Generalization performance is examined by comparing the performance of each model on held out test data from the same source as the training data (internal) to its performance on test data from new hospitals (external), where we use receiver-operating characteristic (ROC) curves to quantify performance. The architectures designed specifically for detection of COVID-19 in radiographs include CV19-Net [310], COVID-Net [291], and DarkCovidNet [209]. The additional "off-the-shelf" models include ResNeXT [301] and MobileNet [241]. (Caption continued on next page)

Figure 4.8: **(Previous page.) Generalization performance of models that were specifically designed in previous studies for detection of COVID-19 in chest radiographs as well as additional "off-the-shelf" architectures.** The "augmented DenseNet-121" is the same as our primary DenseNet-121 model with the addition of the data augmentation scheme from CV19-Net; it therefore represents an intermediate between our primary model and CV19-Net, which is an ensemble of twenty of the "augmented DenseNet-121" models, and it is provided to disentangle the effects of the CV19-Net data augmentation scheme from the effects of ensembling. For example, while the data-augmented DenseNet-121 provides a small but insignificant improvement in external test set performance over the same network without data augmentation for one of the two datasets (panel b, external test set AUC of 0.76 ± 0.04 vs. 0.79 ± 0.03 before and after data augmentation, respectively, when trained on dataset I, p = 0.22, $U = 6$ using two-tailed Mann-Whitney U-test; external test set AUC of 0.70 ± 0.05 vs 0.69 ± 0.05 before and after data augmentation, respectively, when trained on dataset II, p = 1.0, $U = 13$ using two-tailed Mann-Whitney U-test), we find no evidence of significant improvement between the ensembled and single DenseNet-121 models for either dataset (panels a and b, external test set AUC of 0.79 ± 0.04 vs. 0.80 ± 0.02 before and after ensembling, respectively, when trained on dataset I, p = 0.5476, $U = 16$ using two-tailed Mann-Whitney U-test; external test set AUC 0.69 ± 0.05 vs. 0.69 ± 0.04 before and after ensembling, respectively, when trained on dataset II, p = 0.84, $U = 11$ using two-tailed Mann-Whitney U-test). Inset values indicate area under the ROC curve (AUC, mean ± standard deviation, n=5).

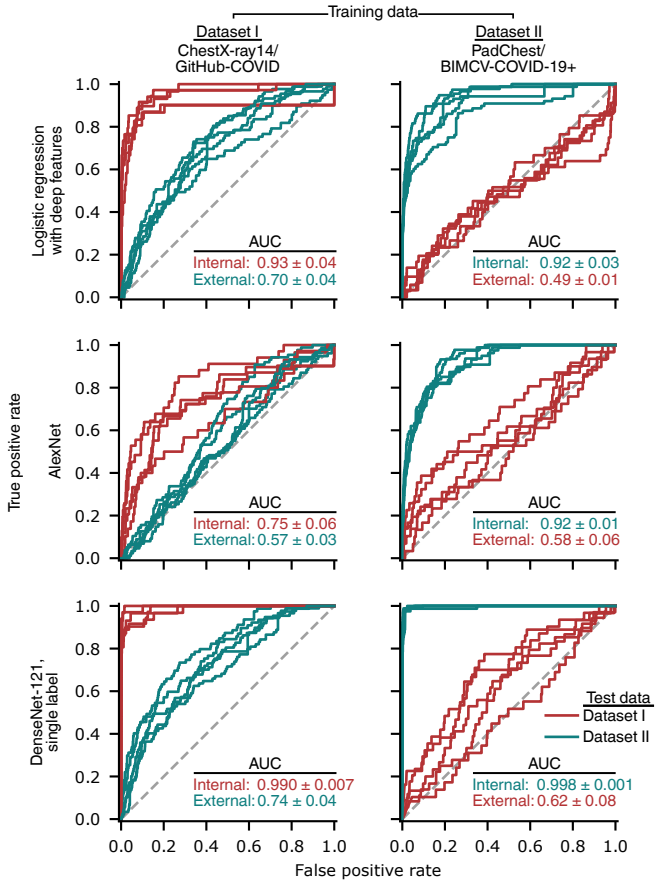

Figure 4.9: **Generalization performance of models with lower capacity or reduced label information, as measured by receiver-operating characteristic (ROC) curves.** The first two rows correspond to models in which the capacity to overfit, which has been implicated in learning of spurious associations [240], has been reduced. The logistic regression with deep features comprises a neural network with the DenseNet-121 architecture that was trained on the ImageNet dataset to derive a set of of 1024 general image features, *i.e.* those output by the penultimate layer of the network, which were used as inputs for a logistic regression; the weights of the neural network were held fixed during training of the logistic regression. The AlexNet models follow the original AlexNet model architecture [142] but with the final 1000-class classification head replaced by a 15-class classification head, corresponding to the 14 ChestX-ray14 labels plus an additional label for COVID-19. The final row represents models with an identical architecture and training scheme to those in the main text, except with only a single output corresponding to presence/absence of COVID-19. Red and teal numbers indicate area under the ROC curves (AUC, mean ± standard deviation, n=5).

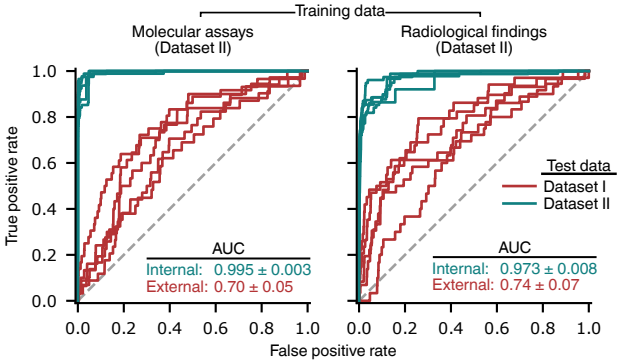

Figure 4.10: **Evaluation of the impact on generalization performance of *concept shift*, a change in the classification task between the training and testing datasets.** In addition to the learning of spurious correlations that do not remain constant between datasets, generalization performance may also drop due to changes in non-spurious correlations between datasets, including a shift in how the labels are generated. In particular, the GitHub-COVID dataset [43], which consists largely of radiographs published in academic articles, may predominantly feature COVID-19+ images with radiological evidence of COVID-19, while COVID-19 labels for the BIMCV-COVID-19+ dataset [288] may be derived from molecular assays (left panel), including reverse-transcription polymerase chain reaction and serology, or from a radiologist's assessment for radiological evidence of COVID-19 (right panel) in addition to confirmation by molecular assays. Specifically, we defined "radiological evidence of COVID-19" as presence of *COVID-19* or *COVID-19 uncertain* in the radiologist-derived labels of BIMCV-COVID-19+. In the event that poor generalization performance is due to a shift from predicting presence of COVID-19, with or without radiological evidence, in the training data, to predicting radiological evidence of COVID-19 in the test data, generalization performance would be expected to increase substantially. Red and teal numbers indicate area under the ROC curves (AUC, mean ± standard deviation, n=5).

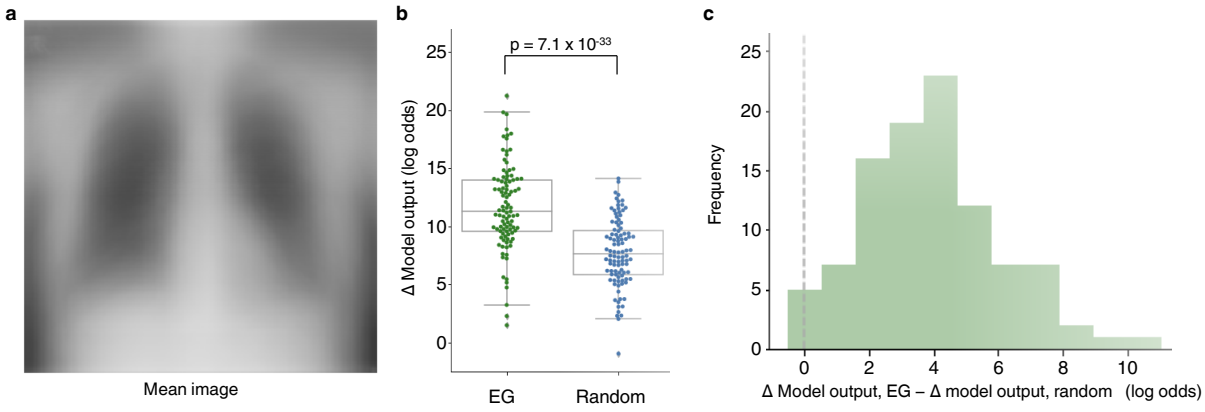

Figure 4.11: **Ablation tests to assess the importance of pixels that are highlighted by saliency maps. a**, Average image of COVID-19+ radiographs from dataset I, from which pixels are drawn to "ablate", *i.e.*, hide, putatively important parts of individual radiographs in our experiment. **b**, Comparison of the change in an AI-based COVID-19 classification model's predictions when pixels are ablated based on their saliency map importance scores or by random. For a randomly chosen subset of radiographs, the 10% of pixels with the highest magnitude expected gradients (EG) scores were ablated by replacing those pixels with the corresponding pixels from the average COVID-19+ image, and as a control, an equivalent number of pixels were replaced at random. Note that in both cases, the model's predicted log odds that the radiograph represents a COVID-19+ patient is expected to increase, since pixels are replaced with pixels from the mean COVID-19+ image. The boxes mark the quartiles (25th, 50th, and 75th percentiles) of the distribution, while the whiskers extend to show the minimum and maximum of the distribution (excluding outliers). Each boxplot marks the 25th, 50th, The p-value is calculated by a two-sided Wilcoxon signed-rank test, n=100 (Siegel's T statistic = 7.69, $p = 1.48 \times 10^{-14}$. **c**, Pairwise comparison of the change in the model's predictions, to assess the superiority of EG relative to random choice at determining important pixels. Since the potential for ablation to change the model's prediction varies from image to image, overlap in the distributions of "EG" and "random" in **b** does *not* imply that for any given image random choice is superior to EG. If for any image a random choice of pixels were superior to EG at determining important pixels, we would expect to observe values less than zero in the histogram, which shows image-level, pairwise differences between EG and random choice.

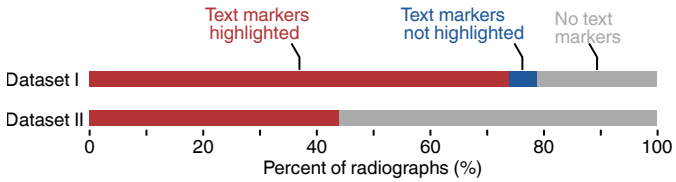

Figure 4.12: **Analysis of the frequency at which saliency maps highlight laterality markers as important features.** To assess the frequency, a random sample of 100 radiographs and their corresponding saliency maps was chosen from each dataset, and each radiograph was manually categorized as (i) contains a laterality marker that is highlighted by the saliency map, (ii) contains a laterality marker that is not highlighted by the saliency map, or (iii) does not contain a laterality marker.

Figure 4.13: **Saliency maps for 15 radiographs from the PadChest, BIMCV-COVID-19+, and ChestX-ray14 repositories.** Across the data sources, saliency maps highlight text tokens and laterality markers (e.g., the first radiograph-saliency map pair in the first row of the PadChest examples, the second-to-last and last radiograph-saliency map pairs in the third row of the PadChest examples, the first four radiograph-saliency map pairs in the second row of the BIMCV examples, and all five radiograph-saliency map pairs in the third row of the ChestX-ray14 examples). For a version of this figure that includes example attributions for the GitHub-COVID repository, see our GitHub repository at `https://github.com/suinleelab/cxr_covid`.

Figure 4.14: **Example images generated by a CycleGAN that was trained to alter COVID-19 negative images from the ChestX-ray14 dataset to appear like COVID-19 positive images from the GitHub-COVID dataset and vice versa.** See our GitHub repository at `https://github.com/suinleelab/cxr_covid` for a version of this figure that includes images from the GitHub-COVID repository.

Figure 4.15: **Examples images generated by a CycleGAN that was trained to alter COVID-19 negative images from the PadChest dataset to appear like COVID-19 positive images from the BIMCV-COVID-19+ dataset and vice versa. (See arXiv Document, Removed Here For Space).**

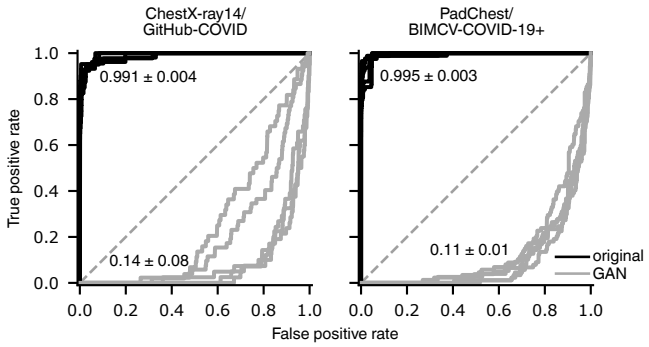

Figure 4.16: **Evaluation of the extent to which features relied upon by the COVID-19 detection models are altered by the CycleGAN, as measured by the drop in classification performance following transformation by the CycleGAN.** A CycleGAN that more reliably alters images such that they appear to the classifier to be of the COVID-19 label opposite their original will achieve an area under the ROC curve (AUC) closer to zero. Inset values indicate AUC (mean ± standard deviation, n=5).

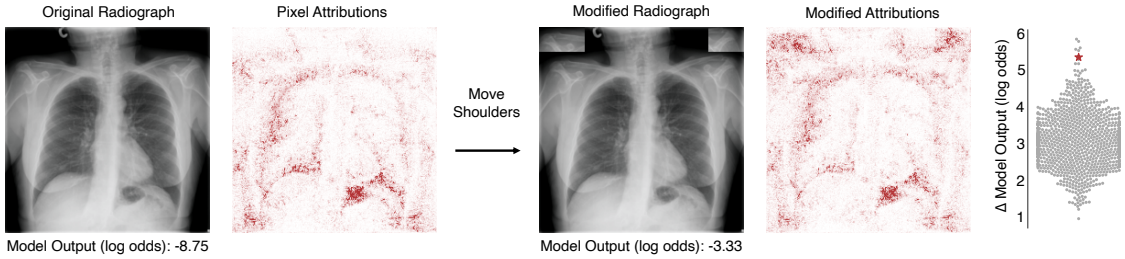

Figure 4.17: **Additional assessment of the importance of shoulder positioning to an AI model for radiographic COVID-19 detection.** The procedure to generate Figure 2d was replicated with a new radiograph; *i.e.*, a patch of the radiograph containing the patient's clavicles was copied to the top corners of the image, and the increase in the model's predicted log odds of COVID-19 was compared to that produced by copying random image patches of the same size ($\Delta = 5.42$, empircal p-value $= 7 \times 10^{-3}$ based on Monte Carlo substitution of random image patches, n=1000) (see Methods Section 2.5).

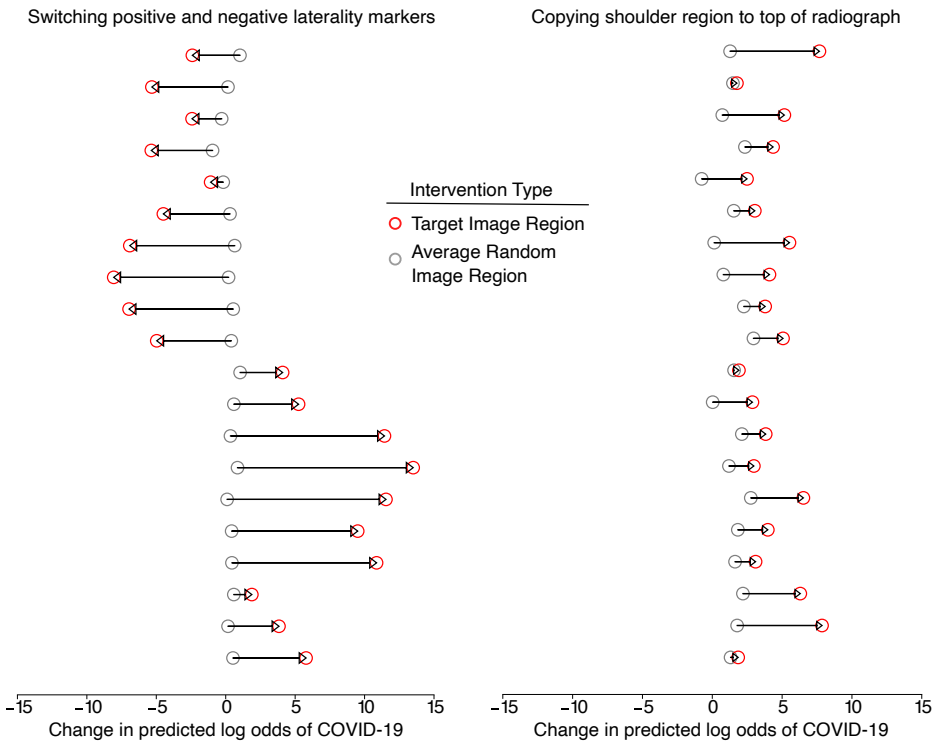

Figure 4.18: **Population-level analysis of importance of laterality markers and shoulder positioning.** Each pair of dots corresponds to an radiograph sampled at random from the larger population, which enables inference of our findings to the population level, despite the infeasibility of completing these experiments for the complete dataset (Dataset II). In each pair, the red dot indicates the difference between the model's predicted log odds of COVID-19 following a targeted intervention on the region of interest and the model's predicted log odds of COVID-19 for the original, unaltered image. The gray dot provides a negative control by repeating the intervention with 1000 random, rather than targeted, image patches of the same size, and then taking the average over the resulting set of changes in the model output. In the left panel, the targeted intervention is to replace the laterality marker on a radiograph from the COVID-19+ repository with a laterality marker on a radiograph from the COVID-19− repository (top 10 radiographs) or vice versa (bottom 10 radiographs), while the untargeted intervention is to swap random image patches of the same size. (Caption continued on next page).

Figure 4.18: **(Previous page.) Population-level analysis of importance of laterality markers and shoulder positioning.** In the experiments in the left panel, radiographs were sampled at random from the subset with laterality markers. In the right panel, the targeted intervention is to copy the shoulder region of the radiograph and move it to the top of the image, while the untargeted intervention is to copy a random region of the same dimensions as the targeted intervention and move it to a random position. In the experiments in the right panel, radiographs were sampled at random from the full set of images. Swapping of laterality markers between COVID-19+ and COVID-19− radiographs produces a significantly greater change in model output than swapping random image patches ($p=8.9 \times 10^{-5}$, Siegel's T statistic = 0.0, by two-tailed Wilcoxon signed rank test, $n=20$ random radiographs), and similarly, movement of the shoulder regions to the top of the radiograph produces a significantly greater change in model output than moving random image patches of the same size ($p=8.9 \times 10^{-5}$, Siegel's T statistic = 0.0 by two-tailed Wilcoxon signed rank test, $n=20$ random radiographs).

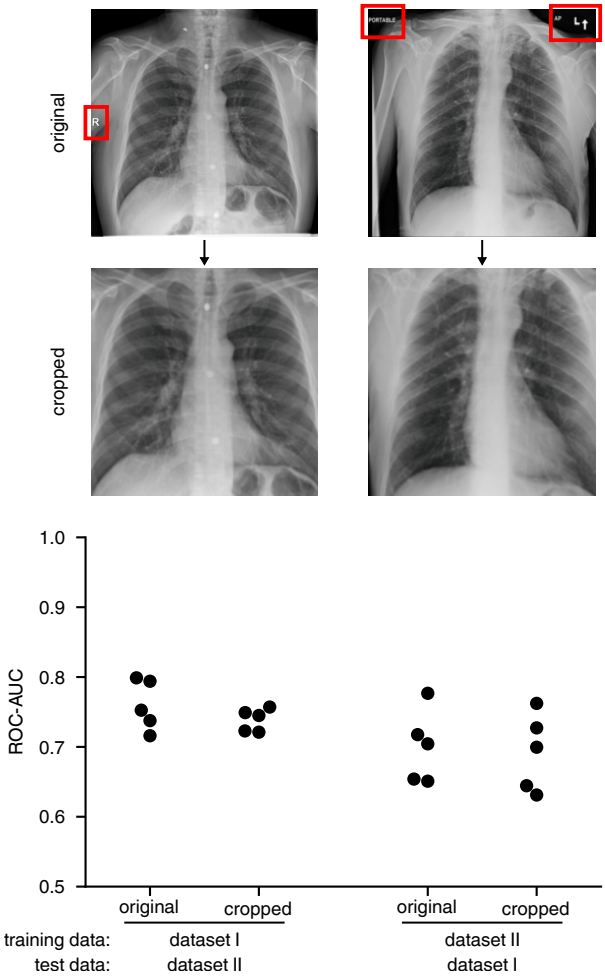

Figure 4.19: **Evaluation of the extent to which image cropping mitigates shortcut learning.** For each dataset, models were trained before and after cropping to the center 75% of the radiograph, which removes from the edge of radiographs the textual markers (red boxes) that may contribute to shortcut learning. Models were then evaluated on an external test set, consisting of radiographs from a different hospital than the training data, to evaluate the generalization performance. Cropping of images did not significantly improve generalization performance based on a one-tailed signed-rank test, where the alternative hypothesis is that the median ROC-AUC of the model trained on cropped images is greater than that trained on the original images (p=0.46 and p=0.60 for models trained on datasets I and II, respectively, based on the Mann-Whitney U-test; corresponding test statistics are U=0.73 and U=0.52, respectively ; n=5 independently trained models).

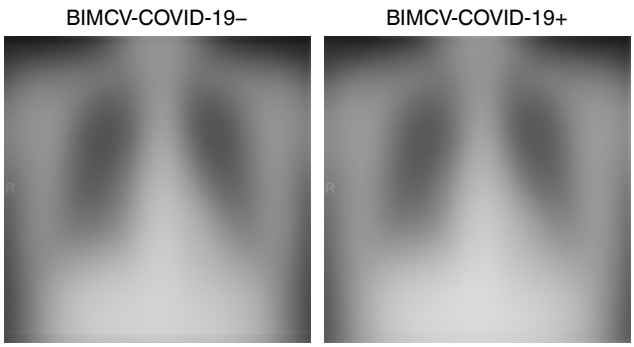

Figure 4.20: **Average images of the BIMCV-COVID-19– and BIMCV-COVID-19+ repositories.** Note consistency in the laterality markers, shoulder positioning, and radiopacity of image borders.

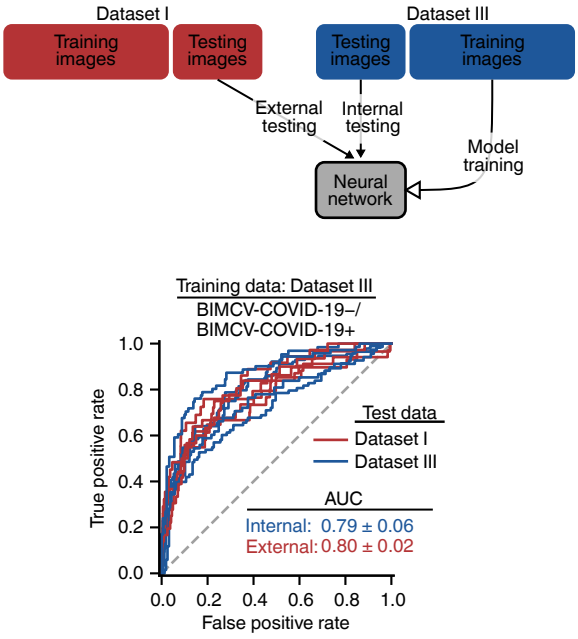

Figure 4.21: **Evaluation of the generalization performance of models trained on dataset III, via ROC curves.** Models are evaluated on both an internal test set (new, held-out examples from the same data source as the training radiographs), and an external test set (radiographs from a new hospital system). Inset numbers indicate the area under the ROC curves (AUC, mean ± standard deviation), where larger area corresponds to higher performance. The difference between internal and external test set performance is the generalization gap.

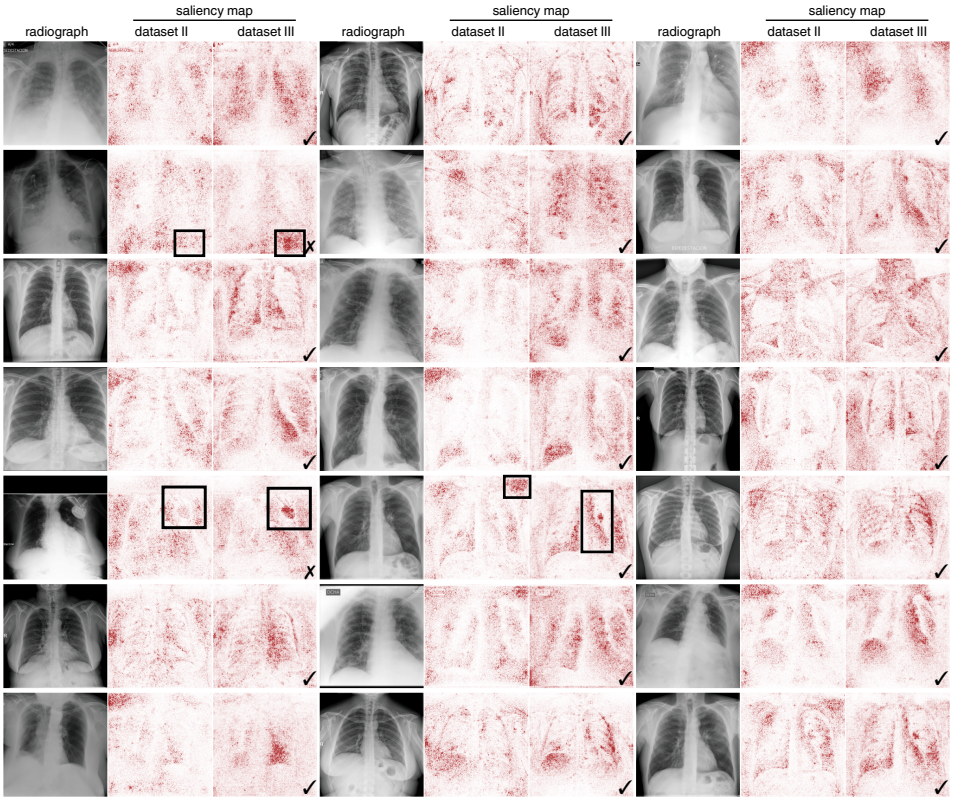

Figure 4.22: **Evaluation of the extent to which improved training data mitigates shortcut learning, evaluated by comparison of saliency maps for models trained on dataset II and dataset III.** For a set of images randomly chosen from the BIMCV-COVID-19+ repository, saliency maps were generated for models trained on Dataset II and models trained on Dataset III, which we expect to contain fewer image factors that spuriously enable COVID-19 positive and COVID-19 negative radiographs to be distinguished. As a basic validation, a model that focuses less on shortcuts would be expected to exhibit saliency maps with increased emphasis on the lung fields and decreased emphasis on the image edges; radiographs for which we judged, on this basis, that the model exhibits less dependence on shortcuts when trained on dataset III than dataset II are marked with a check mark, while radiographs that exhibit greater dependence are marked with an "x". The saliency maps of the two radiographs (out of 21) that did not show improvement exhibit increased attention toward a gastric bubble (black boxes, row two) and a medical device (black boxes; row 5, column 1). While gastrointestinal symptoms are sometimes associated with COVID-19 [90], we were unable to identify reports of an association between gastric bubbles and COVID-19, and therefore judged that this factor likely represents a spurious confound. We additionally annotate an example in which the model exhibits increased attention toward relevant factors (black boxes; row 5, column 2), namely a decrease in attention toward the region above the patient's left shoulder, and an increase in attention toward the left perihilar region.

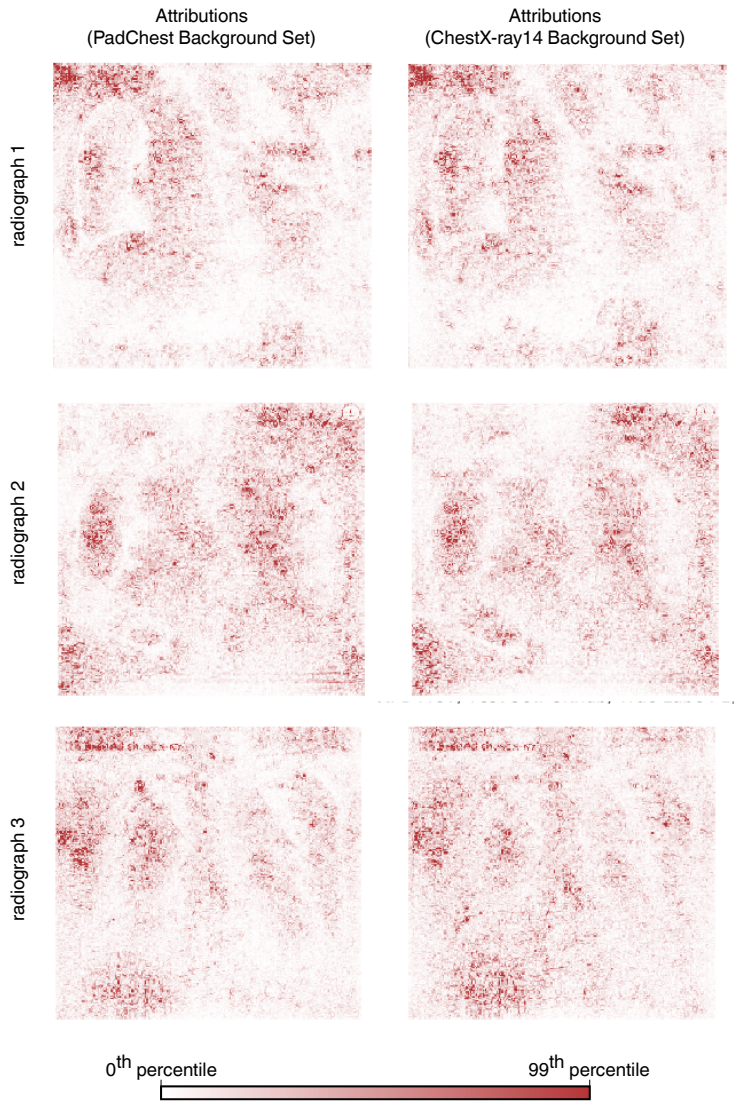

Figure 4.23: **Comparison of expected gradients saliency maps generated from varied reference distributions, which provide the baseline radiographs from which the expected gradients algorithm integrates.**

Chapter 5

IMPROVING PERFORMANCE OF DEEP LEARNING MODELS WITH AXIOMATIC ATTRIBUTION PRIORS AND EXPECTED GRADIENTS

5.1 Introduction

Recent work on interpreting machine learning (ML) models focuses on *feature attribution methods*. Given an input datum, a model, and a prediction, such methods assign a number to each input feature that represents how important the feature was for making the prediction. Current research also investigates the axioms that attribution methods should satisfy [179, 274, 267, 50] and how they provide insight into model behavior [177, 180, 243, 305]. Feature attribution methods often reveal problems in a model or dataset. For example, a model may place too much importance on undesirable features, rely on many features when sparsity is desired, or be sensitive to high frequency noise. In such cases, humans often have a prior belief about how a model should treat input features but find it difficult to mathematically encode this prior for neural networks in terms of the model parameters.

One method to address such problems is what we call an *attribution prior*: if it is possible for explanations to reveal problems in a model, then constraining the model's explanations during training can help the model avoid such problems. It is worth noting that the vast majority of feature attribution methods focus exclusively on explaining *why* a given prediction was made. Only a very small number of papers have investigated incorporating attributions themselves into model training. The first such paper, by Ross et al. [238], used a binary indicator of whether each feature should or should not be important for making predictions on each sample in the dataset and penalized the gradients of unimportant features. A very recent publication successfully uses Ross et al's gradient-based prior as part of a human-in-the-loop strategy to improve model generalization performance and user trust, as well as contributing their own model-agnostic method for penalizing feature importances [246]. Such results create a clear synergy with our study, which improves the quality of calculated feature importances and develops new forms of attribution priors. This has the potential to greatly expand both the number of ways that a human-in-the-loop can influence deep models and the precision with which they can do so. However, two drawbacks limit this method's applicability to real-world problems. First, gradients do not satisfy the same theoretical guarantees as modern feature attribution methods. This leads to well-known problems such as saturation: operations,

like ReLUs and sigmoids, which have large flat "saturated" regions, can lead to 0 gradient attribution even for important features [274]. Second, it can be difficult to specify which features should be important in a binary manner.

Additional recent work discusses the need for priors that incorporate human intuition in order to develop robust and interpretable models [108]. Still, it remains challenging to encode priors such as "have smoother attributions across an image" or "treat this group of features similarly" by penalizing a model's input gradients or parameters. Some recent attribution priors have proposed regularizing integrated gradients (IG) attributions [166, 38]. While promising, this work suffers from three major weaknesses: it does not clearly demonstrate improvements over gradient-based attribution priors, it penalizes attribution deviation from a target value rather than encoding sophisticated priors such as those we mention above, and it imposes a large computational cost by training with tens to hundreds of reference samples per batch. A contemporary method called contextual decomposition explanation penalization (CDEP) uses a framework similar to attribution priors and penalizes explanations generated by the contextual decomposition (CD) method [232]. Unlike all other interpretability methods discussed in this paper, CDEP penalizes explanations for pre-specified *groups of features*, meaning it is best suited for a different set of problems than we consider. More discussion of CDEP can be found in 5.2.1.

The main contribution of this work is a broadened interpretation of attribution priors that includes any case in which the training objective incorporates differentiable functions of a model's feature attributions. This can be seen as a generalization of gradient-based regularization [155, 238, 303, 111, 239] and it can be used to encode meaningful domain knowledge more effectively than existing methods. Whereas previous attribution priors generally took the form of "encourage feature i's attribution to be near a pre-determined target value," the priors we present here consider relative importance among *multiple* features and do not require pre-determined target values for any feature's attribution. Specifically, we introduce an *image prior* enforcing that neighboring pixels have similar attributions, a *graph prior* for biological data enforcing that related genes have similar attributions, and a *sparsity prior* enforcing that a few features have large attributions while all others have near-zero attributions.

We also introduce a new general-purpose feature attribution method to enforce these priors, *expected gradients* (EG). As mentioned above, virtually all attribution methods are designed to explain a model's prediction to humans, not to be penalized during training. This means many such methods may be computationally difficult to incorporate into the training process. EG is the first attribution method explicitly designed for regularization as an attribution prior (Figure 5.1a); it can be efficiently regularized during training due to its formulation as an expectation, which naturally lends itself to batched estimates of the attribution. It also eliminates a hyperparameter choice required by IG [274]. Since these

attributions are used not only to interpret trained models, but also as part of the training objective itself, it is essential to guarantee that the attributions will be of high quality. We therefore show that our attribution method satisfies important interpretability axioms.

Across three different prediction tasks, we show that training with EG outperforms training with previous, more limited versions of attribution priors. On images, our image prior produces a model that is more interpretable and generalizes better to noisy data. On gene expression data, our graph prior reduces prediction error and better captures biological signal. Finally, on a patient mortality prediction task, our sparsity prior yields a sparser model and improves performance when learning from limited training data.

5.2 Results

5.2.1 Attribution priors are a flexible framework for encoding domain knowledge.

Let $X \in \mathbb{R}^{n \times p}$ denote a dataset with labels $y \in \mathbb{R}^{n \times o}$, where n is the number of samples, p is the number of features, and o is the number of outputs. In standard deep learning, we find optimal parameters θ by minimizing loss, with a regularization term $\Omega'(\theta)$ weighted by λ' on the parameters:

$$\theta = \text{argmin}_\theta \mathcal{L}(\theta; X, y) + \lambda' \Omega'(\theta).$$

Attribution priors involve a model's attributions, represented by the matrix $\Phi(\theta, X)$, where each entry ϕ_i^ℓ is the importance of feature i in the model's output for sample ℓ. The attribution prior is a scalar-valued penalty function of the feature attributions $\Omega(\Phi(\theta, X))$, which represents a log-transformed prior probability distribution over possible attributions (λ is the regularization strength). The attribution prior is modular and agnostic to the particular attribution method. This results in the optimization:

$$\theta = \text{argmin}_\theta \mathcal{L}(\theta; X, y) + \lambda \Omega(\Phi(\theta, X)),$$

where the standard regularization term has simply been replaced with an arbitrary, differentiable penalty function on the feature attributions.

While feature attributions have previously been used in training (more details in 5.4) [238, 166], our approach offers two novel components. First, we demonstrate that calculating Φ with attribution methods that satisfy previously-established *interpretability axioms* improves performance (see Section 2.2 and 5.5 for further discussion of interpretability axioms). Second, rather than simply encouraging each feature's attribution to be near a target value as in previous work, we enforce *high-level* priors over the relationships between features.

In image data, we use a Laplace 0-mean prior on the difference between attributions of adjacent pixels, which encourages a low total variation (high smoothness) of attributions:

$$\Omega_{\text{pixel}}(\Phi(\theta, X)) = \sum_{\ell} \sum_{i,j} |\phi^{\ell}_{i+1,j} - \phi^{\ell}_{i,j}| + |\phi^{\ell}_{i,j+1} - \phi^{\ell}_{i,j}|,$$

where i, j indexes the pixels of an image by rows and columns, respectively and ℓ indexes each image.

In gene expression data, we use a Gaussian 0-mean prior on the difference between mean absolute attributions $\bar{\phi}_i$ of functionally related genes, which encourages such similar genes to have similar attributions:

$$\Omega_{\text{graph}}(\Phi(\theta, X)) = \sum_{i,j} W_{i,j}(\bar{\phi}_i - \bar{\phi}_j)^2 = \bar{\phi}^T L_G \bar{\phi},$$

where $W_{i,j}$ is the weight of connection between two genes in a biological graph, and L_G is the graph Laplacian.

Finally, in health data where sparsity is desired, we use a prior on the Gini coefficient of the mean absolute attributions $\bar{\phi}_i$, which encourages a small number of features to have a large percentage of the total attribution while others are near-zero:

$$\Omega_{\text{sparse}}(\Phi(\theta, X)) = -\frac{\sum_{i=1}^{p} \sum_{j=1}^{p} |\bar{\phi}_i - \bar{\phi}_j|}{n \sum_{i=1}^{p} \bar{\phi}_i} = -2G(\bar{\phi}),$$

where G is the Gini coefficient.

None of these priors require specifying target values for features, and all improve performance over simpler baselines. For more details on our priors see Section 5.6, and for more details on previous attribution priors, see Section 5.4. We also note that these priors involve the relationships between the attributions for all features in the dataset. Gradients, IG, and our method (EG) discussed below are all designed for calculating such attributions. The CDEP method discussed above differs in that it penalizes the attributions of a single pre-specified group of features [232]; while CDEP has reported better performance with certain types of priors than EG and gradients, we believe this is due to the fact that the methods are inherently best suited to different types of priors. Using CDEP with the specific priors proposed in this work would require several orders of magnitude more backward passes of the model during training than our approach. CDEP also uses additional preprocessing steps which are not necessary in our approach, which further distinguishes the scenarios in which each method is most applicable.

5.2.2 Expected gradients outperforms other attribution methods.

Attribution priors involve using feature attributions not just as a post-hoc analysis method, but as a key part of the training objective. Thus, it is essential to guarantee as much as possible

that the attribution method used will produce high-quality attributions and run fast enough to be calculated for each training batch. We propose an axiomatic feature attribution method called *expected gradients* (EG), which avoids problems with existing methods and is naturally suited to being incorporated into training. EG extends the integrated gradients method [274], and like IG, satisfies a variety of desirable interpretability axioms such as completeness (the feature attributions sum to the output for a given sample) and implementation invariance (the attributions are identical for any of the infinite possible implementations of the same function). Because these methods satisfy completeness, they are not subject to the problems with input saturation that affect gradient attributions. Because these methods satisfy implementation invariance, they are straight-forward to practically apply to any differentiable model, regardless of specific network architectures (see Section 5.5 for an extended discussion of the interpretability axioms satisfied by EG).

Integrated gradients generates feature attributions by integrating the gradients of the model's output between the sample of interest and a *reference* sample x' (5.1a, left).

$$\text{IntegratedGradients}_i(x) := \int_{\alpha=0}^{1} \frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha$$

If the attribution function Φ in our attribution prior $\Omega(\Phi(\theta, X))$ is integrated gradients, regularizing Φ would require hundreds of extra gradient calls every training step (the original IG paper [274] recommends 20 to 300 gradient calls to compute attributions). This makes training with IG prohibitively slow – in fact, [166] find that using IG can take up to 30 times longer than standard training even when only back-propagating gradients through part of the network. However, most deep learning models today are trained using some variant of batch gradient descent, where the gradient of a loss function is approximated over many training steps using mini-batches of data. We can dramatically improve speed over an IG attribution prior by using a similar idea and formulating the IG integral as an expectation (see Table 5.1 for more details on convergence time benchmark): this Monte Carlo estimate of the integral is the core of our *expected gradients* method, defined below for a single reference x':

$$\text{SingleRefEG}_i(x) = \mathop{\mathbb{E}}_{\alpha \sim U(0,1)} \left[(x_i - x'_i) \times \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \right]$$

Just like the gradient of the loss, EG attributions can be calculated in a batched manner during training (5.1a, right). We let k be the number of samples we draw for this Monte Carlo integral at each mini-batch. Remarkably, because the variance in each batched EG attribution will be smoothed over thousands of batches during training, we find that as small as $k = 1$ suffices to regularize the explanations.

This expectation formulation also enables us to solve a longstanding problem with integrated gradients as an *attribution* method – the choice of the required background reference x'. For example, in image tasks, the image of all zeros is often chosen as a baseline, but doing so implies that black pixels will not be highlighted as important (5.1b and 5.1c). This problem can be solved by integrating gradients over multiple references. However, calculating multiple Riemann integrals is expensive in terms of time and memory, likely prohibitively so if calculated during every batch of training (5.1a, right). EG naturally accommodates multiple references by performing the Monte Carlo integral with samples from multiple references *and* interpolation points (here, x is the sample, x' is a reference, and D is the reference distribution):

$$\text{ExpectedGradients}_i(x) = \mathop{\mathbb{E}}_{x' \sim D, \alpha \sim U(0,1)} \left[(x_i - x'_i) \times \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \right]$$

In principle, any distribution D over reference samples could be used to calculate EG attributions; choosing which distribution to use depends on the nature of the attribution problem. For example, setting D to be a single sample recovers single-reference EG: the same reference setup as IG but with the Monte Carlo speedup of EG. By default, we do not choose D to be a single sample but rather a uniform distribution over the entire training set. This tells us which features cause x's output to be different from the output at all other points in the dataset, on average. In certain cases we may want to use a different distribution D. For example, we might want to distinguish between subgroups and understand why a digit is classified as a "seven" rather than a "one" by choosing references only from the "one"-labeled training samples. We could also account for baseline subgroup characteristics by explaining, for example, an 80-year-old patient's mortality risk relative to other 80-year-olds; this could prevent age and age-correlated features from being trivially listed as the most important. While our formulation and implementation of EG support any choice of distribution D, the examples in this paper do not focus on subgroup analysis, so we set D to be a uniform distribution over the training set (see Section 5.5).

In a simple experiment using synthetic data to assess the impact of k on the convergence time of model training (rather than the convergence of a single explanation), we found that regularizing EG with $k = 1$ was more effective at removing a model's dependency on one of two correlated features than gradients or even IG with more than k samples (see Table 5.1). The $k = 1$ setting also appeared optimal for EG; setting $k > 1$ required more total gradient calls for convergence. We also compare EG to other feature attribution methods using synthetic data benchmarks introduced in [178] (Table 5.1), which are available as part of the SHAP software package. These benchmark metrics evaluate whether each attribution method finds the most important features for a given dataset and model. EG significantly

Table 5.1: **Synthetic data benchmark results for attribution methods.** Larger numbers mean a better feature attribution method for all metrics other than Convergence Time, for which a smaller number indicates faster convergence. The first three metrics measure the quality of the method for correctly identifying important features, while convergence time indicates how effectively the method is regularized during training as an attribution prior. The "Remove Positive" metric measures the average magnitude change in model output when the features identified as having the largest *positive* impact by each method are masked by the feature mean, while "Remove Negative" measures the average magnitude change in model output when the features identified as having the largest *negative* impact by each method are masked by the feature mean. The "Remove Absolute" metric measures the average increase in model loss when the features identified as having the largest magnitude impact on the model are masked by the feature mean. Each model is trained on 900 samples and tested using 100 samples. EG attains the best benchmark scores of all of the tested attribution methods ($p = 7.2 \times 10^{-5}$, one-tailed Binomial test, tested across all 18 attribution performance metrics.

Method	Remove Positive	Remove Negative	Remove Absolute	Convergence Time
Expected Grad.	**3.612**	**3.759**	**0.897**	**0.150**
Integrated Grad.	3.539	3.687	0.872	0.989
Gradients	0.035	0.110	0.729	0.250
Random	-0.053	0.034	0.400	—

outperforms the next best feature attribution method ($p = 7.2 \times 10^{-5}$, one-tailed Binomial test). We believe this demonstrates another benefit of EG; by averaging attributions over multiple reference samples, it becomes more robust to the wide array of patterns of missingness and re-imputation tested in the benchmark.

5.2.3 A pixel attribution prior improves robustness to image noise.

Prior work on interpreting image models focused on creating *pixel attribution maps*, which assign a value to each pixel indicating how important that pixel was for a model's prediction [250, 274]. Attribution maps can be noisy and difficult to understand due to their tendency to highlight seemingly unimportant background pixels, indicating the model may be vulnerable to adversarial attacks [237]. Although we may prefer a model with smoother attributions, existing methods only post-process attribution maps but do not change model behavior [262, 250, 73]. Such techniques may not be faithful to the original model [108]. In this section, we describe how we applied our framework to train image models with naturally smoother

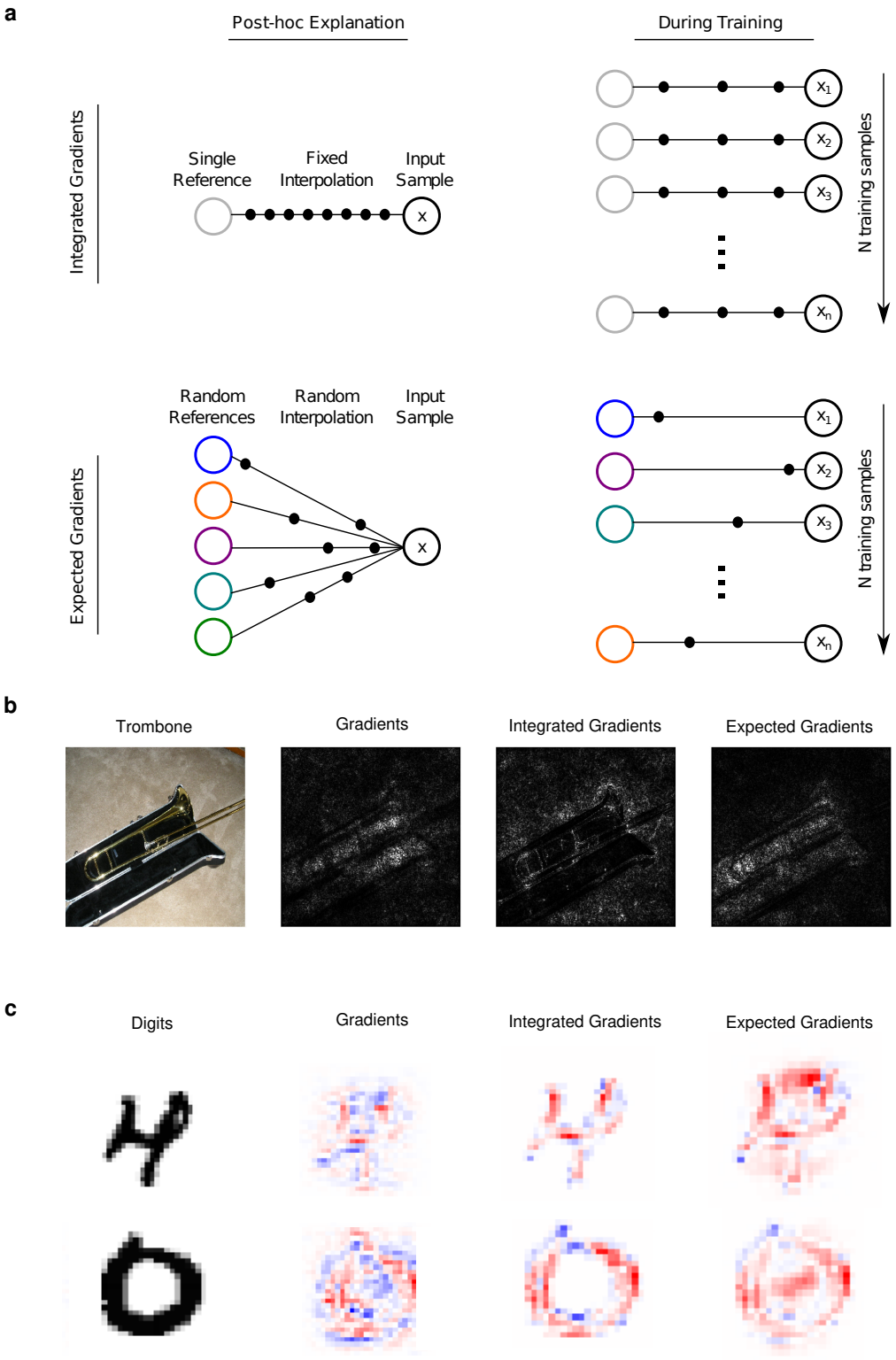

Figure 5.1: **Expected Gradients is a feature attribution method designed to be regularized during training.** (Caption continued on next page.)

Figure 5.1: (Previous page.) **a**, A comparison of our method, expected gradients (EG), to integrated gradients (IG) as both a post-hoc explanation method (left), and as a differentiable feature attribution to be penalized during training to enforce attribution priors (right). **b**, Comparison of saliency maps generated by three different attribution methods on an image from the ImageNet dataset. The saliency maps demonstrate how the IG attribution method fails to highlight black pixels as important when black is used as a baseline input, while EG is capable of highlighting the black pixels in these images as important. **c**, Comparison of saliency maps for the same three attribution methods for two MNIST digits. Again, IG fails to highlight potentially relevant image regions (like the empty middle of the 0 or the empty region at the top of the 4 which might make the digit resemble a 9 if it were filled in).

attributions.

To regularize pixel-level attributions, we used the following intuition: neighboring pixels should have a similar impact on an image model's output. To encode this intuition, we chose a total variation loss on pixel-level attributions (see Section 5.6 for more detail). We applied this pixel smoothness attribution prior to the MNIST and CIFAR-10 datasets [155, 141]. On MNIST we trained a two-layer convolutional neural network; for CIFAR-10 we trained a VGG16 network from scratch (see Section 5.7 for more details) [259]. In both cases we optimized hyperparameters for the baseline model without an attribution prior. To choose λ, we searched over values in $[10^{-20}, 10^{-1}]$ and chose the λ that minimized the attribution prior penalty and achieved a test accuracy within 1% of the baseline model for MNIST and 10% for CIFAR-10. Figures 5.2 and 5.3 display EG attribution maps for both the baseline and the model regularized with an attribution prior on 5 randomly selected test images on MNIST and CIFAR-10, respectively. In all examples, the attribution prior yields a model with visually smoother attributions. Remarkably, in many instances smoother attributions better highlight the target object's structure.

Recent work has suggested that image classifiers are brittle to small domain shifts: small changes in the underlying distribution of the training and test set can significantly reduce test accuracy [228]. To simulate a domain shift, we applied Gaussian noise to images in the test set and re-evaluated the performance of the regularized and baseline models. As an adaptation of [238], we also compared the attribution prior model to regularizing the total variation of gradients with the same criteria for choosing λ. For each method, we trained 5 models with different random initializations. In Figures 5.2 and 5.3, we plot the mean and standard deviation of test accuracy on MNIST and CIFAR-10, respectively, as a function of standard deviation of added Gaussian noise. The figures show that our regularized model is

more robust to noise than both the baseline and gradient-based models.

Both the robustness and more intuitive saliency maps our method provides come at the cost of reduced test set accuracy (0.93 ± 0.002 for the baseline vs. 0.85 ± 0.003 for pixel attribution prior model on CIFAR-10). Mathematically, adding a penalty term to the optimization objective should only ever reduce training set performance; it is reasonable that in many cases this can lead to a reduction in test-set performance as well. However, test accuracy is not the only metric of interest for image clasifiers. The trade-off between robustness and accuracy that we observe is consistent with previous work that suggests image classifiers trained solely to maximize test accuracy rely on features that are brittle and difficult to interpret [108, 285, 307]. Despite this trade-off, we find that at a stricter hyperparameter cutoff for λ on CIFAR-10 – within 1% test accuracy of the baseline, rather than 10% – our methods still achieve modest but significant robustness relative to the baseline. We also evaluated our method against several other attribution priors including IG and, for ablation purposes, single-reference EG. We found that the pixel attribution prior outperformed standard IG and that most of this additional performance was due to our random interpolation. Both the pixel attribution prior and single-reference EG were much more robust than all other methods; however, only the pixel attribution prior, which used multiple references, could highlight important foreground *and* background regions in addition to providing robustness and smoothness.

5.2.4 A Graph attribution prior improves anti-cancer drug response prediction.

In the image domain, our attribution prior took the form of a penalty encouraging smoothness over adjacent pixels. In other domains, there may be prior information about specific relationships between features that can be encoded as a graph (such as social networks, knowledge graphs, or protein-protein interactions). For example, prior work in bioinformatics has shown that protein-protein interaction networks contain valuable information for improving performance on biological prediction tasks [40]. Therefore, in this domain we regularized attributions to be smooth over the protein-protein feature graph analogously to the regular graph of pixels in the image.

Incorporating the Ω_{graph} attribution prior not only led to a model with more reasonable attributions but also improved predictive performance by letting us incorporate prior biological knowledge into the training process. We downloaded publicly available gene expression and drug response data for patients with acute myeloid leukemia (AML, a type of blood cancer) and tried to predict patients' drug response from their gene expression [286]. For this regression task, an input sample was a patient's gene expression profile plus a one-hot encoded vector indicating which drug was tested in that patient, while the label we tried to predict was drug response (measured by IC50, a continuous value representing the concentration of

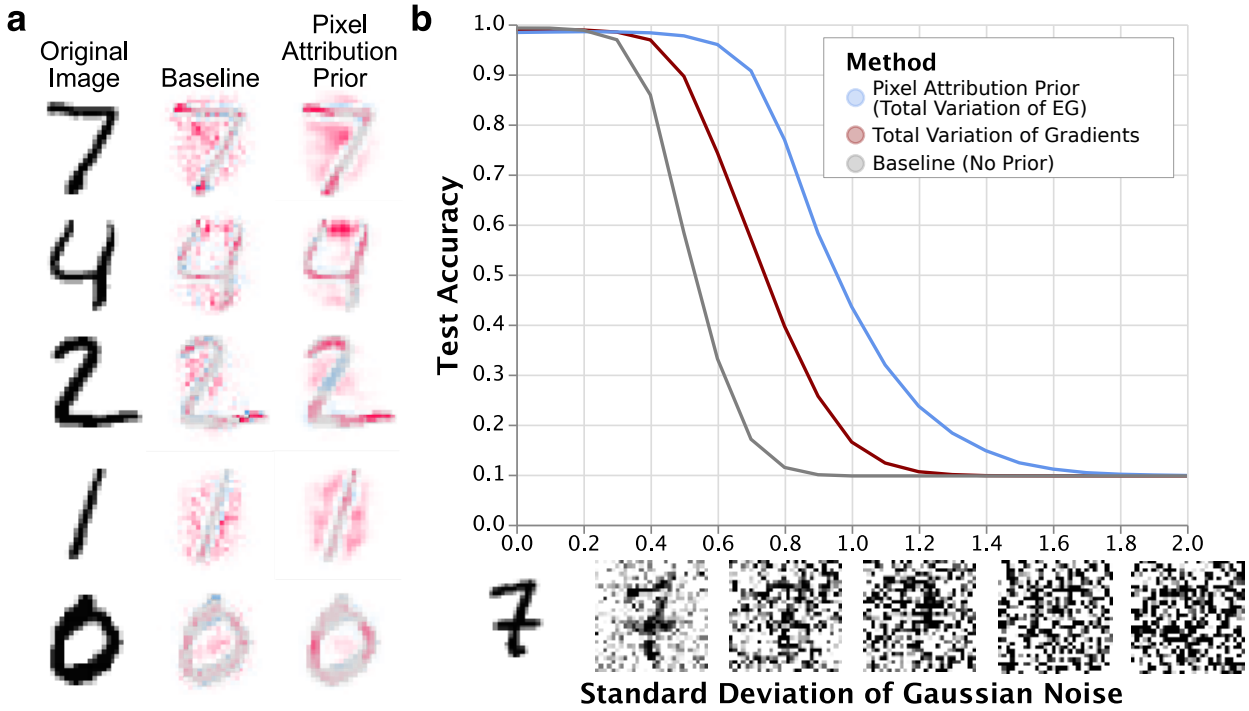

Figure 5.2: **Pixel Attribution Prior improves saliency map smoothness and increases robustness of MNIST classifier to noise.** **a**, EG attributions (from 100 samples) on MNIST for both an unregularized model and a model trained with an attribution prior regularized using EG. The latter achieves visually smoother attributions, and it better highlights how the network classifies digits (e.g., the top part of the 4 being very important). Unlike previous methods which take additional steps to smooth saliency maps after training [262, 73], these are *unmodified* saliency maps directly from the learned model. **b**, Training with an attribution prior on total variance of EG attributions induces robustness to Gaussian noise without specifically training for robustness. This robustness greatly exceeds that provided by an attribution prior on the total variance of model gradients. Shaded bars around each line indicate standard deviation of the accuracy results; however, the bars are small enough to be indistinguishable in this plot.

the drug required to kill half of the patient's tumor cells). To define the graph used by our prior, we downloaded the tissue-specific gene interaction graph for the tissue most closely related to AML in the HumanBase database [88].

A two-layer neural network trained with our graph attribution prior (Ω_{graph}) significantly outperforms all other methods in terms of test set performance as measured by R^2, which

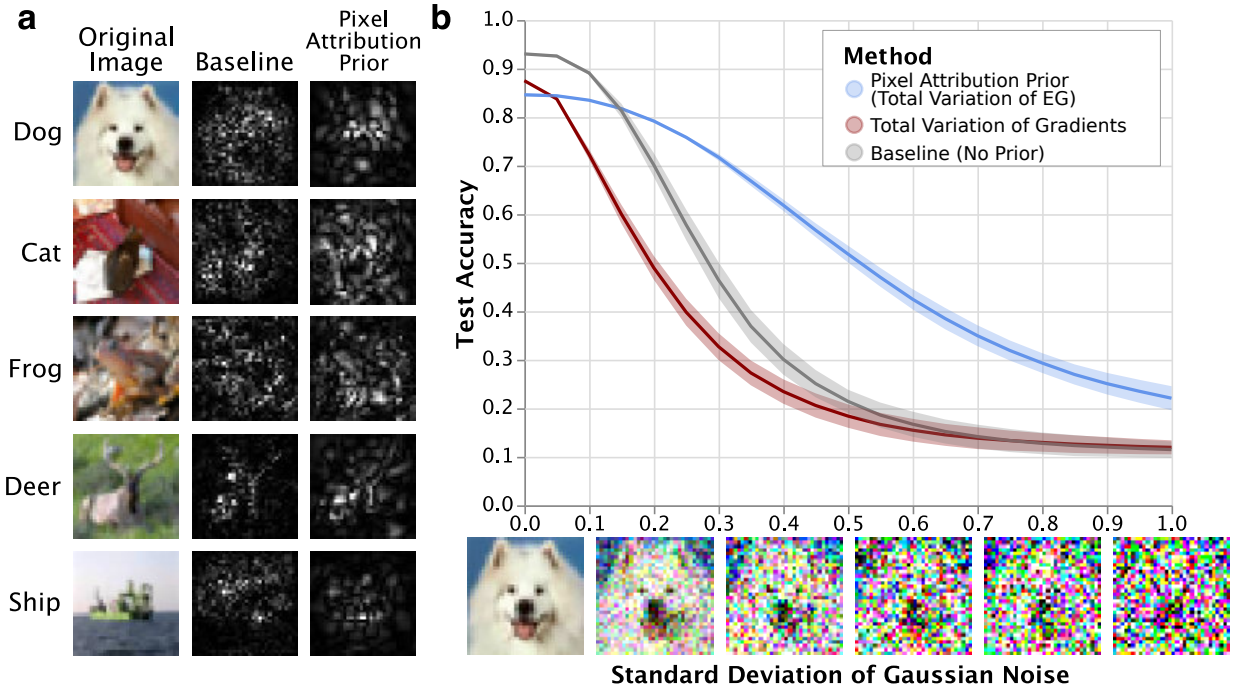

Figure 5.3: **Pixel Attribution Prior improves saliency map smoothness and increases robustness of CIFAR10 classifier to noise. a**, EG attributions (from 100 samples) on CIFAR10 for both the baseline model and the model trained with an attribution prior for five randomly selected images classified correctly by both models. Training with an attribution prior generates visually smoother attribution maps in all cases. Notably, these smoothed attributions also appear more localized towards the object of interest. **b**, Training with an attribution prior on total variance of EG attributions induces robustness to Gaussian noise, achieving more than double the accuracy of the baseline at high noise levels. This robustness is not achievable by choosing total variation of gradients as the attribution function. Shaded bars around each line indicate standard deviation of the accuracy results.

indicates the fraction of the variance in the output explained by the model (Figure 5.4, see Section 5.8 for significance testing). Unsurprisingly, when we replace the biological graph from HumanBase with a randomized graph, we find that the test performance is no better than the performance of a neural network trained without *any* attribution prior. Extending the method proposed in [238] by applying our new graph prior as a penalty on the model's *gradients*, rather than a penalty on the axiomatically correct expected gradient feature attribution, does not perform significantly better than a baseline neural network. We also observe substantially improved test performance when using the prior graph information to regularize a linear LASSO model. Finally, we note that our graph attribution prior neural network significantly outperforms graph convolutional neural networks, a recent method for utilizing graph information in deep neural networks [132].

To find out if our model's attributions match biological domain knowledge, we first compared the list of top genes generated by our network trained with a graph attribution prior (ranked by mean absolute feature attribution) to a "ground truth" list of AML-relevant genes found by querying the GeneCards database (5.4b). When we count the number of AML-relevant genes at each position in our network's top gene list and compare this to the number of AML-relevant genes at each position in a standard neural network's top gene list, we see that the graph attribution prior network captures significantly more biologically-relevant genes.

Additionally, to check for biological pathway-level enrichments, we conducted Gene Set Enrichment Analysis (a modified Kolmogorov–Smirnov test). We measured whether our top genes, ranked by mean absolute feature attribution, were enriched for membership in any pathways (see Section 5.8 for more detail, including the top pathways for each model) [270]. We find that the neural network with the tissue-specific graph attribution prior captures far more biologically-relevant pathways (increased number of significant pathways after FDR correction) than a neural network without attribution priors [18]. Furthermore, the pathways our model uses more closely match biological expert knowledge, i.e., they included prognostically useful AML gene expression profiles as well as important AML-related transcription factors [167]. These results are expected, given that neural networks trained without priors can learn a relatively sparse basis of genes that will not enrich for specific pathways (e.g. a single gene from each correlated pathway), while those trained with our graph prior will spread credit among functionally-related genes. This demonstrates the graph prior's value as an accurate and efficient way to encourage neural networks to treat functionally-related genes similarly.

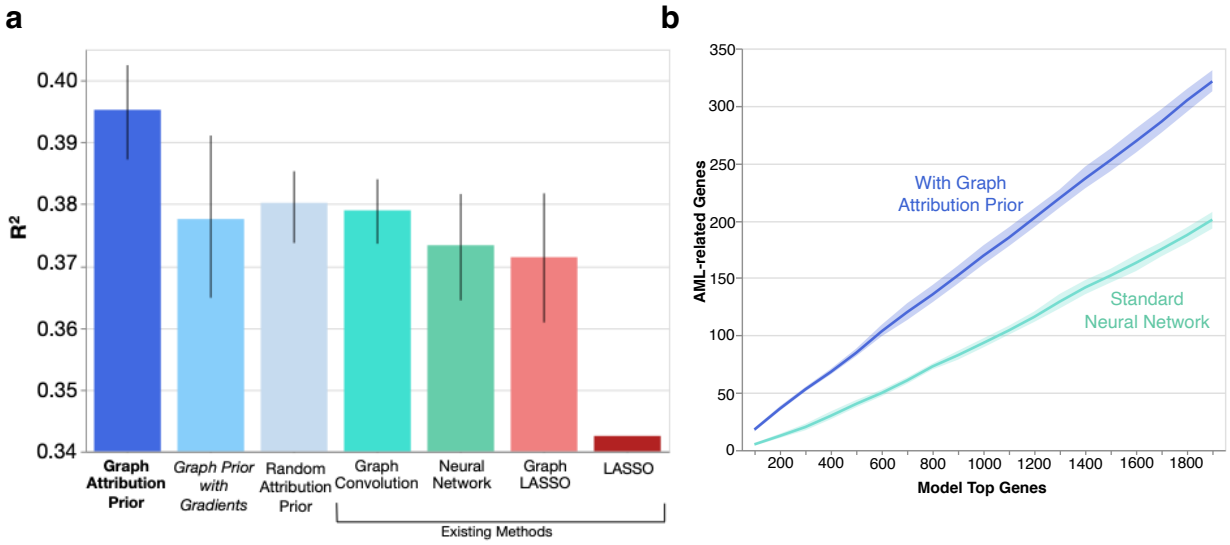

Figure 5.4: **Graph Attribution Prior improves test accuracy and biological relevance of anti-cancer drug response prediction model.** **a**, A neural network trained with our graph attribution prior (bold) attains the best test performance, while one trained with the same graph penalty on the gradients (italics, adapted from [238]) does not perform significantly better than a standard neural network (error bars indicate the extent of the bootstrapped 95% confidence interval of the mean test set R^2 value, over 10 re-trainings of the model on random re-splits of the data.). **b**, A neural network trained with our graph attribution prior gives more weight to AML-relevant genes than a standard neural network trained without the graph attribution prior (solid line indicates average over 10 random re-splits of the data and re-trainings of the model, error bands indicate the extent of the bootstrapped 95% confidence interval).

5.2.5 A sparsity prior improves performance with limited training data.

Feature selection and *sparsity* are popular ways to alleviate the curse of dimensionality, facilitate interpretability, and improve generalization by building models that use a small number of input features. A straightforward way to build a sparse deep model is to apply an L1 penalty to the first layer (and possibly subsequent layers) of the network. Similarly, the Sparse Group Lasso (SGL) method penalizes all weights connected to a given feature [70, 244], while a simple existing attribution prior approach [236] penalizes the gradients of each feature in the model.

These approaches suffer from two problems. First, a feature with small gradients or first-layer weights may still strongly affect the model's output [256]. A feature whose attribution value (e.g., integrated or expected gradients) is zero is much less likely to have any effect on predictions. Second, successfully minimizing penalties like L1 – regardless of attribution type – is not necessarily the best way to create a sparse model. A model that puts weight w on 1 feature is penalized more than one that puts weight $\frac{w}{2p}$ on each of p features. Prior work on sparse linear regression has shown that the Gini coefficient G of the weights, proportional to 0.5 minus the area under the CDF of sorted values, avoids such problems and corresponds more directly to a sparse model [105, 314]. We extend this analysis to deep models by noting that the Gini coefficient can be written differentiably and used as an attribution prior.

Here, we show that the Ω_{sparse} attribution prior can build sparser models that perform better in settings with limited training data. We use a publicly available healthcare mortality prediction dataset of 13,000 patients [191], whose 35 features (118 after one-hot encoding) represent medical data such as a patient's age, vital signs, and laboratory measurements. The binary outcome is survival after 10 years. Sparse models in this setting may enable accurate models to be trained with very few labeled patient samples or reduce cost by accurately risk-stratifying patients using few lab tests. We randomly sampled training and validation sets of only 100 patients each, placing all other patients in the test set, and ran each experiment 200 times with a new random sample to average out variance. We built 3-layer binary classifier neural networks regularized using L1, SGL, and sparse attribution prior penalties to predict patient survival, as well as an L1 penalty on gradients adapted for global sparsity from [238, 236]. The regularization strength was tuned from 10^{-7} to 10^{5} using the validation set for all methods (see Section 5.9).

The sparse attribution prior enables more accurate test predictions (Figure 5.5a) and sparser models (Figure 5.5c) when limited training data is available, with $p < 10^{-4}$ and $T \geq 4.314$ by paired-samples T-test for all comparisons. We also plot the average cumulative importance of sorted features and find that the sparse attribution prior more effectively concentrates importance in the top few features (Figure 5.5d). In particular, we observe that L1 penalizing the model's gradients as in [236] rather than its EG attributions performs

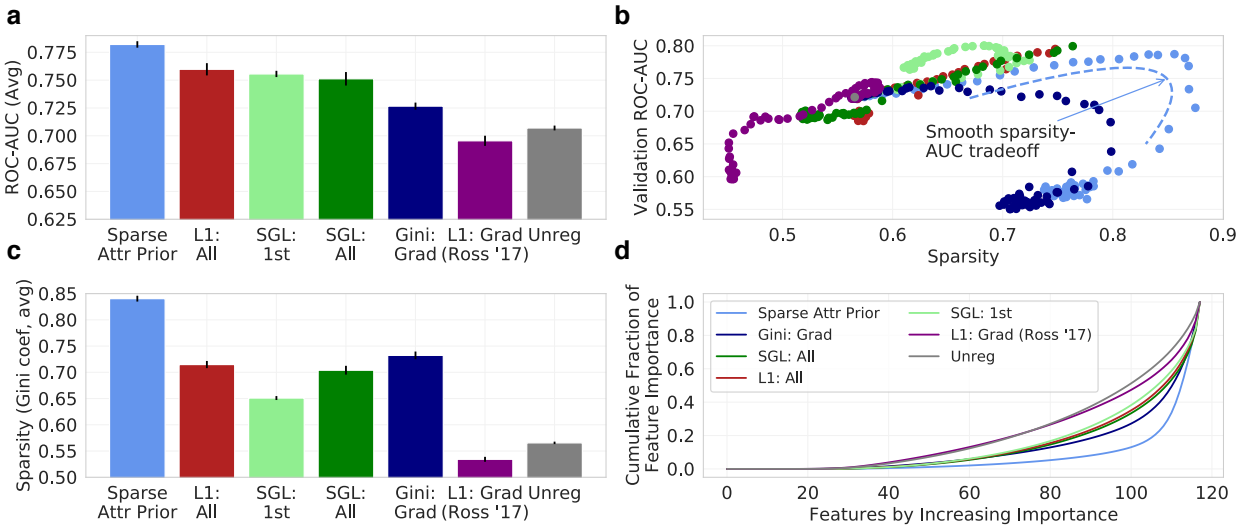

Figure 5.5: **Sparse Attribution Prior builds sparser and more accurate healthcare mortality models.** A sparse attribution prior enables more accurate test predictions (**a**) and sparser models (**c**) across 200 small subsampled datasets (100 training and 100 validation samples, all other samples used for test set) than other penalties, including gradients. **b**, Across the full range of tuned parameters, the sparse attribution prior achieves the greatest sparsity and a smooth sparsity-validation performance trade-off. **d**, A sparse attribution prior concentrates a larger fraction of global feature importance in the top few features. "Gini", "L1", and "SGL" indicate the Gini, L1, and SGL penalties respectively. "Grad" indicates a penalty on the gradients, "All" indicates a penalty on all weights in the model, and "1st" indicates a penalty on only the first weight layer.

poorly in terms of both sparsity and performance. A Gini penalty on gradients improves sparsity but does not outperform other baselines like SGL and L1 in ROC-AUC. Finally, we plot the average sparsity of the models (Gini coefficient) against their validation ROC-AUC across the full range of regularization strengths. The sparse attribution prior exhibits higher sparsity than other models and a smooth tradeoff between sparsity and ROC-AUC (Figure 5.5b).

5.3 Discussion

The immense popularity of deep learning has driven its application in many areas with diverse, complicated domain knowledge. While it is in principle possible to hand-design network architectures to encode this knowledge, a more practical approach involves the use of

attribution priors, which penalize the importance a model places on each of its input features when making predictions. Unfortunately, previous attribution priors have been limited, both theoretically and computationally. Binary penalties only specify whether features should or should not be important and fail to capture relationships among features. Approaches that only focus on a model's input gradients change the local decision boundary but often fail to impact a model's underlying decision-making. Attribution priors on more complicated attributions, like integrated gradients, have proven computationally difficult.

Our work advances previous work both by introducing novel, flexible attribution priors for multiple domains and by enabling the training of such priors with a newly defined feature attribution method. Our priors lead to smoother and more interpretable image models, biological predictive models that incorporate graph-based prior knowledge, and sparser healthcare models that perform better in data-scarce scenarios. Our attribution method not only enables the training of said priors, but also outperforms its predecessor – integrated gradients – in terms of reliably identifying the features models use to make predictions.

There remain many avenues for future work in this area. We chose to base our prior on an improved version of integrated gradients because it is the most prominent differentiable feature attribution method we are aware of, but a wide array of other attribution methods exist. Our framework makes it straightforward to substitute any other attribution method as long as it is differentiable, and studying the effectiveness of other attribution methods as priors would be valuable. In addition, while we develop new, more sophisticated attribution priors and show their value, there is ample room to improve on our priors and evaluate entirely new ones for other tasks. Determining the best attribution priors for particular tasks opens a further avenue of research. We believe that surveys of domain experts to establish model desiderata for particular applications will help to develop the best priors for any given situation while offering a valuable opportunity to put humans in the loop. Overall, the dual advances of sophisticated attribution priors and expected gradients enable a broader view of attribution priors: as tools to achieve domain-specific goals without sacrificing efficiency.

5.4 Previous attribution priors

The first instance of what we now call an attribution prior was proposed in [238], where the regularization term was modified to place a constant penalty on the gradients of undesirable features:

$$\theta = \text{argmin}_\theta \mathcal{L}(\theta; X, y) + \lambda'' ||A \odot \frac{\partial \mathcal{L}}{\partial X}||_F^2.$$

Here, the attribution method is the gradients of the model, represented by the matrix $\frac{\partial \mathcal{L}}{\partial X}$ whose ℓ, ith entry is the gradient of the loss at the ℓth sample with respect to the ith feature.

A is a binary matrix indicating which features should be penalized in which samples.

A more general interpretation of attribution priors is that *any function* of *any feature attribution method* could be used to penalize a loss function, thus encoding prior knowledge about what properties the attributions of a model should have. For some model parameters θ, let $\Phi(\theta, X)$ be a feature attribution method, which is a function of θ and the data X. Let ϕ_i^ℓ be the feature importance of feature i in sample ℓ. We formally define an *attribution prior* as a scalar-valued penalty function of the feature attributions $\Omega(\Phi(\theta, X))$, which represents a log-transformed prior probability distribution over possible attributions:

$$\theta = \text{argmin}_\theta \mathcal{L}(\theta; X, y) + \lambda\Omega(\Phi(\theta, X)),$$

where λ is the regularization strength. Note that the attribution prior function Ω is agnostic to the attribution method Φ.

Previous attribution priors [238, 166] required specifying an exact target value for the model's attributions, but often we do not know in advance which features are important in advance. In general, there is no requirement that $\Phi(\theta, X)$ constrain attributions to particular values. Section 5.2 presented three newly developed attribution priors for different tasks that improve performance without requiring pre-specified attribution targets for any particular feature.

5.5 Expected gradients

Expected gradients is an extension of integrated gradients [274] with fewer hyperparameter choices. Like several other attribution methods, integrated gradients aims to explain the difference between a model's current prediction and the prediction that the model would make when given a baseline input. This baseline input is meant to represent some uninformative reference input that represents not knowing the value of the input features. Although choosing such an input is necessary for several feature attribution methods [274, 256, 21], the choice is often made arbitrarily. For example, for image tasks, the image of all zeros is often chosen as a baseline, but doing so implies that black pixels will not be highlighted as important by existing feature attribution methods. In many domains, it is not clear how to choose a baseline that correctly represents a lack of information.

Our method avoids an arbitrary choice of baseline; it models not knowing the value of a feature by integrating over a dataset. For a model f, the *integrated gradients* value for feature i is defined as:

$$\text{IntegratedGradients}_i(x, x') := (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha,$$

where x is the target input and x' is baseline input. To avoid specifying x', we define the *expected gradients* value for feature i as:

$$\text{ExpectedGradients}_i(x) := \int_{x'} \left((x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha \right) p_D(x') dx',$$

where D is the underlying data distribution. Since EG is also a diagonal path method, it satisfies the same axioms as IG [74]. Directly integrating over the training distribution is intractable; therefore, we instead reformulate the integrals as expectations:

$$\text{ExpectedGradients}_i(x) := \mathop{\mathbb{E}}_{x' \sim D, \alpha \sim U(0,1)} \left[(x_i - x'_i) \times \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \right].$$

This expectation-based formulation lends itself to a natural, sampling based approximation method: (1) draw samples of x' from the training dataset and α from $U(0, 1)$, (2) compute the value inside the expectation for each sample and (3) average over samples.

EG also satisfies a set of important interpretability axioms: implementation invariance, sensitivity, completeness, linearity, and symmetry-preserving.

- *Implementation invariance* states that two networks with outputs that are equal over all inputs should have equivalent attributions. Any attribution method based on the gradients of a network will satisfy this axiom [274], meaning that IG, EG, and gradients will all be implementation invariant.

- *Sensitivity* (sometimes called Dummy) states that when a model does not depend on a feature at all, it receives zero importance. IG, EG, and gradients all satisfy sensitivity because the gradient w.r.t. an irrelevant feature will be 0 everywhere.

- *Completeness* states that the attributions should sum to the difference between the output of a function at the input to be explained and the output of that function at a baseline. Gradients do *not* satisfy completeness due to saturation at the inputs; elements like ReLUs may cause gradients to be zero, making completeness impossible [274]. IG and EG both satisfy completeness due to the gradient theorem (fundamental theorem of calculus for line integrals) [274]. For EG, the function being integrated is the expectation of the model's output, so completeness means that the attributions sum to the difference between the model's output for the input and the model's output averaged over all possible baselines.

- *Linearity* states that for a model that is a linear combination of two submodels $f(x) = af_1(x) + bf_2(x)$, the attributions are a linear combination of the submodels'

attributions $\phi(x) = a\phi_1(x) + b\phi_2(x)$. This will hold for IG, EG, and gradients because gradients are linear.

- *Symmetry-preserving* states that symmetric variables with identical values should achieve identical attributions. IG is symmetry preserving since it is a straight line path method, and EG will also be symmetry preserving, as a symmetric function of symmetric functions will itself be symmetrical [274].

Unlike previous attribution methods, EG is explicitly designed for natural batched training. This enables an order of magnitude increase in computational efficiency relative to previous approaches for training with attribution priors. We further improve performance by reducing the need for additional data reading. Specifically, for each input in a batch of inputs, we need k additional inputs to calculate EG attributions for that input batch. As long as k is smaller than the batch size, we can avoid any additional data reading by re-using the same batch of input data as a reference batch, as in [308]. We accomplish this by shifting the batch of input k times, such that each input in the batch uses k other inputs from the batch as its reference values.

5.6 Specific priors

Here, we elaborate on the explicit form of the attribution priors we used in this paper. In general, minimizing the error of a model corresponds to maximizing the likelihood of the data under a generative model consisting of the learned model plus parametric noise. For example, minimizing mean squared error in a regression task corresponds to maximizing the likelihood of the data under the learned model, assuming Gaussian-distributed errors:

$$\arg\min_\theta ||f_\theta(X) - y||_2^2 = \arg\max_\theta \exp(-||f_\theta(X) - y||_2^2) = \theta_{MLE},$$

where θ_{MLE} is the maximum-likelihood estimate of θ under the model $Y = f_\theta(X) + \mathcal{N}(0, \sigma)$.

An additive regularization term is equivalent to adding a multiplicative (independent) prior to yield a maximum a posteriori estimate:

$$\arg\min_\theta ||f_\theta(X) - y||_2^2 + \lambda||\theta||_2^2 = \arg\max_\theta \exp(-||f_\theta(X) - y||_2^2)\exp(-\lambda||\theta||_2^2) = \theta_{MAP},$$

Here, adding an L2 penalty is equivalent to MAP for $Y = f_\theta(X) + \mathcal{N}(0, \sigma)$ with a $\mathcal{N}(0, \frac{1}{\lambda})$ prior. We next discuss the functional form of the attribution priors enforced by our penalties.

5.6.1 Pixel attribution prior

Our pixel attribution prior is based on the anisotropic total variation loss and is given as follows:

$$\Omega_{\text{pixel}}(\Phi(\theta, X)) = \sum_{\ell} \sum_{i,j} |\phi^{\ell}_{i+1,j} - \phi^{\ell}_{i,j}| + |\phi^{\ell}_{i,j+1} - \phi^{\ell}_{i,j}|,$$

where $\phi^{\ell}_{i,j}$ is the attribution for the i, j-th pixel in the ℓ-th training image. Research shows [13] that this penalty is equivalent to placing 0-mean, iid, Laplace-distributed priors on the differences between adjacent pixel values, i.e., $\phi^{\ell}_{i+1,j} - \phi^{\ell}_{i,j} \sim \text{Laplace}(0, \lambda^{-1})$ and $\phi^{\ell}_{i,j+1} - \phi^{\ell}_{i,j} \sim \text{Laplace}(0, \lambda^{-1})$. [13] does not call our penalty "total variation," but it is in fact the widely used anisotropic version of total variation and is directly implemented in Tensorflow [2, 172, 254].

5.6.2 Graph attribution prior

For our graph attribution prior, we used a protein-protein or gene-gene interaction network and represented these networks as a weighted, undirected graph. Formally, assume we have a weighted adjacency matrix $W \in \mathbb{R}^{p \times p}_{+}$ for an undirected graph, where the entries encode our prior belief about the pairwise similarity of the importances between two features. For a biological network, $W_{i,j}$ encodes either the probability or strength of interaction between the i-th and j-th genes (or proteins). We encouraged similarity along graph edges by penalizing the squared Euclidean distance between each pair of feature attributions in proportion to how similar we believe them to be. Using the graph Laplacian ($L_G = D - W$), where D is the diagonal degree matrix of the weighted graph, this becomes:

$$\Omega_{\text{graph}}(\Phi(\theta, X)) = \sum_{i,j} W_{i,j}(\bar{\phi}_i - \bar{\phi}_j)^2 = \bar{\phi}^T L_G \bar{\phi}.$$

In this case, we choose to penalize *global* rather than local feature attributions. We define $\bar{\phi}_i$ to be the importance of feature i across all samples in our dataset, where this global attribution is calculated as the average magnitude of the feature attribution across all samples: $\bar{\phi}_i = \frac{1}{n} \sum_{\ell=1}^{n} |\phi^{\ell}_i|$. Just as the image penalty is equivalent to placing a Laplace prior on adjacent pixels in a regular graph, the graph penalty Ω_{graph} is equivalent to placing a Gaussian prior on adjacent features in an arbitrary graph with Laplacian L_G [13].

5.6.3 Sparse attribution prior

Our sparsity prior uses the Gini coefficient G as a penalty, which is written:

$$\Omega_{\text{sparse}}(\Phi(\theta, X)) = -\frac{\sum_{i=1}^{p}\sum_{j=1}^{p}|\bar{\phi}_i - \bar{\phi}_j|}{n\sum_{i=1}^{p}\bar{\phi}_i} = -2G(\bar{\phi}),$$

By taking exponentials of this function, we find that minimizing the sparsity regularizer is equivalent to maximizing likelihood under a prior proportional to the following:

$$\prod_{i=1}^{p}\prod_{j=1}^{p}\exp\left(\frac{1}{\sum_{i=1}^{p}\bar{\phi}_i}|\bar{\phi}_i - \bar{\phi}_j|\right),$$

To our knowledge, this prior does not directly correspond to a named distribution. However, we observe that its maximum value occurs when one $\bar{\phi}_i$ is 1 and all others are 0, and that its minimum occurs when all $\bar{\phi}_i$ are equal. This is similar to the total variation penalty Ω_{image}, but it is normalized and has a flipped sign to *encourage* differences. The corresponding attribution prior is maximized when global attributions are zero for all but one feature and minimized when attributions are uniform across features.

5.7 Image model experimental settings

We trained a VGG16 model from scratch modified for the CIFAR-10 dataset, containing 60,000 colored 32x32-pixel images divided into 10 categories, as in [168]. To train this network, we used stochastic gradient descent with an initial learning rate of 0.1 and an exponential decay of 0.5 applied every 20 epochs. Additionally, we used a momentum level of 0.9. For augmentation, we shifted each image horizontally and vertically by a pixel shift uniformly drawn from the range [-3, 3], and we randomly rotated each image by an angle uniformly drawn from the range [-15, 15]. We used a batch size of 128. Before training, we normalized the training dataset to have zero mean and unit variance, and standardized the test set with the mean and variance of the training set. We used $k = 1$ background reference samples for our attribution prior while training. When training with attributions over images, we first normalized the per-pixel attribution maps by dividing by the standard deviation before computing the total variation; otherwise, the total variation can be made arbitrarily small without changing model predictions by scaling down the pixel attributions close to 0.

We repeated the same experiment as above on MNIST, which contains 60,000 black-and-white 28x28-pixel images of handwritten digits. We trained a CNN with two convolutional layers and a single hidden layer. The convolutional layers each had 5x5 filters, a stride length of 1, and 32 and 64 filters total. Each convolutional layer was followed by a max pooling layer of size 2 with stride length 2. The hidden layer had 1024 units and a dropout rate of 0.5 during training [265]. Dropout was turned off when calculating the gradients with respect to the attributions. We trained with the Adam optimizer with the default parameters ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) [131]. We trained with an initial learning rate

of 0.0001, with an exponential decay of 0.95 for every epoch, for a total of 60 epochs. For all models, we trained with a batch size of 50 images and used $k = 1$ background reference sample per attribution while training.

5.8 Biological experiments

5.8.1 Significance testing of results

To test the difference in R^2 attained by each method, we used a T-test for the means of two independent samples of scores (as implemented in SciPy) [289]. This is a two-sided test and can be applied to R^2 since R^2 is a linear transformation of mean squared error, which satisfies normality assumptions by the central limit theorem. When we compare the R^2 attained from 10 independent retrainings of the neural network to the R^2 attained from 10 independent retrainings of the attribution prior model, we find that predictive performance is significantly higher for the model with the graph attribution prior (t-statistic = 3.59, $p = 2.06 \times 10^{-3}$).

To ensure that the increased performance in the attribution prior model was due to real biological information, we replaced the gene-interaction graph with a randomized graph (symmetric matrix with identical number of non-zero entries to the real graph, but entries placed in random positions). We then compared the R^2 attained from 10 independent retrainings of a neural network with no graph attribution prior to 10 independent retrainings of an neural network regularized with the random graph and found that test error was not significantly different between these two models (t-statistic = 1.25,p = 0.23). We also compared to graph convolutional neural networks, and found that our network with a graph attribution prior outperformed the graph convolutional neural network (t-statistic = 3.30,$p = 4.0 \times 10^{-3}$). Finally, we compared to an L2 penalty applied uniformly across all attributions, and found that this attribution prior did not significantly increase performance from baseline (t-statistic = 1.7,p0.12).

5.8.2 Train/validation/test set allocation

To increase the number of samples in our dataset, we used as a feature the identity of the drug being tested, rather than one of a number of possible output tasks in a multi-task prediction. This follows from prior literature on training neural networks to predict drug response [220]. This yielded 30,816 samples (covering 218 patients and 145 anti-cancer drugs). Defining a sample as a drug and a patient, however, meant we had to choose carefully how to stratify samples into our train, validation, and test sets. While it is perfectly legitimate in general to randomly stratify samples into these sets, we wanted to specifically focus on how well our model could learn trends from gene expression data that would generalize to new patients. Therefore, we stratified samples at a patient-level rather than at the level of

individual samples (e.g., no samples from any patient in the test set ever appeared in the training set). We split 20% of the total patients into a test set (6,155 samples) and then split 20% of the training data into a validation set for hyperparameter selection (4,709 samples).

5.8.3 Model class implementations and hyperparameters tested

LASSO. We used the scikit-learn implementation of the LASSO [282, 213]. We tested a range of α parameters from 10^{-9} to 1, and we found that the optimal value for α was 10^{-2} by mean squared error on the validation set.

Graph LASSO. For our Graph LASSO, we used the Adam optimizer in TensorFlow [2], with a learning rate of 10^{-5} to optimize the following loss function:

$$\mathcal{L}(w; X, y) = \|Xw - y\|_2^2 + \lambda' \|w\|_1 + \nu' w^T L_G w, \tag{5.1}$$

where $w \in \mathbb{R}^d$ is the weights vector of our linear model and L_G is the graph Laplacian of our HumanBase network [88]. In particular, we downloaded the "Top Edges" version of the hematopoietic stem cell network, which was thresholded to only have non-zero values for pairwise interactions that had a posterior probability greater than 0.1. We used the value of λ' selected as optimal in the regular LASSO model (10^{-2}, which corresponds to the α parameter in scikit-learn) and then tuned over ν' values ranging from 10^{-3} to 100. We found that a value of 10 was optimal according to MSE on the validation set.

Neural networks. We tested a variety of hyperparameter settings and network architectures via validation set performance to choose our best neural networks, including the following feed-forward network architectures (where each element in a list denotes the size of a hidden layer): [512,256], [256,128], [256,256], and [1000,100]. We tested a range of L1 penalties on all of the weights of the network, from 10^{-7} to 10^{-2}. All models attempted to optimize a least squares loss using the Adam optimizer, with learning rates again selected by hyperparameter tuning ranging from 10^{-5} to 10^{-3}. Finally, we implemented an early stopping parameter of 20 rounds to select the number of epochs of training (training was stopped after no improvement on validation error for 20 epochs, and the number of epochs was chosen based on optimal validation set error). We found that the optimal architecture (chosen by lowest validation set error) had two hidden layers of size 512 and 256, an L1 penalty on the weights of 10^{-3} and a learning rate of 10^{-5}. We additionally found that 120 was the optimal number of training epochs.

Attribution prior neural networks. We next applied our attribution prior to the neural networks. First, we tuned networks to the optimal conditions described above. We then added extra epochs of fine-tuning where we ran an alternating minimization of the following objectives:

$$\mathcal{L}(\theta; X, y) = \|f_\theta(X) - y\|_2^2 + \lambda\|\theta\|_1 \tag{5.2}$$

$$\mathcal{L}(\theta; X) = \Omega_{graph}(\Phi(\theta, X)) = \nu\bar{\phi}^T L_G \bar{\phi} \tag{5.3}$$

Following [238], we selected ν to be 100 so that the Ω_{graph} term would initially be equal in magnitude to the least squares and L1 loss terms. We found that 5 extra epochs of tuning were optimal by validation set error. We drew $k = 10$ background samples for our attributions. To test our attribution prior using gradients as the feature attribution method (rather than expected gradients), we followed the exact same procedure, only we replaced $\bar{\phi}$ with the average magnitude of the gradients rather than the expected gradients.

Graph convolutional networks. We followed the implementation of graph convolution described in [132]. The architectures were searched as follows: in every network we first had a single graph convolutional layer (we were limited to one graph convolution layer due to memory constraints on each Nvidia GTX 1080-Ti GPU that we used), followed by two fully connected layers of sizes (512,256), (512,128), or (256,128). We tuned over a wide range of hyperparameters, including L2 penalties on the weights ranging from 10^{-5} to 10^{-2}, L1 penalties on the weights ranging from 10^{-5} to 10^{-2}, learning rates of 10^{-5} to 10^{-3}, and dropout rates ranging from 0.2 to 0.8. We found the optimal hyperparameters based on validation set error were two hidden layers of size 512 and size 256, an L2 penalty on the weights of 10^{-5}, a learning rate of 10^{-5}, and a dropout rate of 0.6. We again used an early stopping parameter and found that 47 epochs was the optimal number.

5.9 Sparsity experiments

5.9.1 Data description and processing

Our sparsity experiments used data from the NHANES I survey [191] and contained 35 variables (expanded to 118 features by one-hot encoding of categorical variables) gathered from 13,000 patients. The measurements included demographic information like age, sex, and BMI as well as physiological measurements like blood, urine, and vital sign measurements. The prediction task was a binary classification of whether the patient was still alive (1) or not (0) 10 years after data were gathered.

Data were mean-imputed and standardized so that each feature had 0 mean and unit variance. For each of the 200 experimental replicates, 100 train and 100 validation points were sampled uniformly at random; all other points were allocated to the test set.

5.9.2 Model

We trained a range of neural networks to predict survival in the NHANES data. The architecture, nonlinearities, and training rounds were all held constant at values that performed well on an unregularized network, and the type and degree of regularization were varied. All models used ReLU activations and a single output with binary cross-entropy loss; in addition, all models ran for 100 epochs with an SGD optimizer with learning rate 0.001 on the size-100 training data. The entire 100-sample training set fit in one batch. Because the training set was so small, all of its 100 samples were used for EG attributions during training and evaluation, yielding $k = 100$. Each model was trained on a single GPU on a desktop workstation with 4 Nvidia 1080 Ti GPUs.

Architecture. We considered a range of architectures, including single-hidden-layer 32-node, 128-node, and 512-node networks, two-layer [128,32] and [512,128]-node networks, and a three-layer [512,128,32]-node network; we fixed the [512,128,32] architecture for future experiments.

Regularizers. We tested a large array of regularizers in addition to those considered in the maintext.

5.9.3 Hyperparameter tuning

We selected the hyperparameters for our models based on validation performance. We searched all L1, L2, SGL and attribution prior penalties with 121 points sampled on a log scale over $[10^{-7}, 10^{5}]$.

5.9.4 Maintext methods

Performance and sparsity bar plots. The performance bar graph (Figure 5.5a) was generated by plotting mean test ROC-AUC of the best model of each type (chosen by validation ROC-AUC) averaged over each of the 200 subsampled datasets, with confidence intervals given by 2 times the standard error over the 200 replicates. The sparsity bar graph (Figure 5.5c) was constructed using the same process, but with Gini coefficients rather than ROC-AUCs.

Feature importance distribution plot. The distribution of feature importances was plotted in the main text as a Lorenz curve (Figure 5.5, bottom right): for each model, the features were sorted by global attribution value $\bar{\phi}_i$, and the cumulative normalized value of the lowest q features was plotted, from 0 at $q = 0$ to 1 at $q = p$. A lower area under the curve indicates more features had relatively small attribution values, indicating the model was sparser. Because 200 replicates were run on small subsampled datasets, the Lorenz curve for each model was plotted using the averaged mean absolute sorted feature importances over

all replicates. Thus, for a given model type, the $q = 1$ point represented the mean absolute feature importance of the least important feature averaged over each replicate, $q = 2$ added the mean importance for the second least important feature averaged over each replicate, and so on.

Performance vs sparsity plot. Validation ROC-AUC and model sparsity were calculated for each of the 121 regularization strengths and averaged over each of the 200 replicates. These were plotted on a scatterplot to show the possible range of model sparsities and ROC-AUC performances (Figure 5.5, top right) as well as the tradeoff between sparsity and performance.

Statistical significance. Statistical significance of the sparse attribution prior performance was assessed by comparing the test ROC-AUCs of the sparse attribution prior models on each of the 200 subsampled datasets to those of the other models (L1 gradients, L1 weights, SGL, and unregularized). Significance was assessed by 2-sided paired-samples T-test, paired by subsampled dataset. The same process was used to calculate the significance of model sparsity as measured by the Gini coefficient.

Code Availability

Implementations of attribution priors for Tensorflow and PyTorch are available at `https://github.com/suinleelab/attributionpriors`. This repository also contains code reproducing main results from the paper.

Data Availability

The data for all experiments and figures in the paper are publicly available. The repository above contains a downloadable version of the dataset used for the sparsity experiment, as well as links to download the datasets used in the image and graph prior experiments. Data for the benchmarks was published as part of [176] and can be accessed at `https://github.com/suinleelab/treeexplainer-study/tree/master/benchmark`

Chapter 6

AN ADVERSARIAL APPROACH FOR THE ROBUST CLASSIFICATION OF PNEUMONIA FROM CHEST RADIOGRAPHS

6.1 Introduction

A variety of recent papers have demonstrated the promise of deep learning for medical imaging tasks. From the prediction of diabetic retinopathy using retinal scan images to the diagnosis of melanoma from photographs, machine learning approaches have achieved near-physician level performance [91, 67]. Deep learning classifiers of chest radiographs are not only promising in a research setting, but have also been deployed in clinical practice. For example, an algorithm to detect 4 different thoracic diseases from frontal chest radiographs was evaluated in an emergency medicine setting and was found to increase radiology residents' sensitivity [106].

Despite these major advances, there are still significant limitations for medical deep learning. One of these problems is dataset shift, or the loss in performance when a model is tested on data that is drawn from a different distribution than the data used for training the model [223, 219]. Zech et al. [305] found that a deep learning pneumonia classifier trained on data from two hospital systems exploited differences in the base rate of pneumonia between the two hospitals by learning to identify each radiograph's hospital of origin rather than anatomically-relevant features of pneumonia. While this model apparently had high predictive performance, when the model was tested on radiographs from a third hospital not present in the training data its performance significantly decreased. Furthermore, even within a single hospital system, confounded predictions may be a problem for deep learning. For example, Badgeley et al. [11] demonstrated that a deep learning hip fracture classifier was leveraging patient-level variables (such as age and gender) and process-level variables (such as scanner model and hospital department) in its predictions. After controlling for these variables during model evaluation by rebalancing the test set, they found that the classifier performed no better than random. A recent multi-society statement on the "Ethics of Artificial Intelligence in Radiology" points to the importance of being able to understand and guide the decision-making process of machine learning algorithms to ensure that these algorithms can be safely and effectively used in clinical practice [80]. While the works above have described the brittleness of deep learning medical imaging classifiers, more work is needed to create robust models.

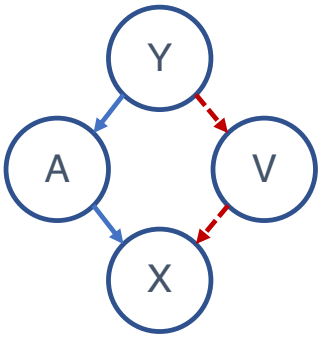

Figure 6.1: Causal graph showing relationships that form part of one plausible data generating process for chest radiographs: relationships are between pneumonia (Y), radiograph view position (V), anatomically relevant radiographic features (A), and the final chest radiograph (X). Red and dashed edges indicate a view-mediated causal path between the radiograph and pneumonia that may shift between different datasets or hospitals. We emphasize that this does not illustrate the full data generating process, and that many data generating processes are possible.

We propose an approach based on adversarial neural networks to address dataset shift by learning models that are invariant to confounders that may shift across hospitals. In particular, we focus on the problem of pneumonia classification from chest radiographs, as the problem of confounding and dataset shift has been particularly well-documented for this task [305]. We find that (1) potential model confounding can be effectively identified by evaluating how well confounders can be predicted from a model's output, that (2) adversarial training enables pneumonia classification that is independent of radiograph view, and that (3) the adversarially-trained models attain better generalization performance when tested in novel hospital systems. [1]

6.2 Problem Statement

We first consider some of the causal relationships forming part of one plausible data generating process for chest radiographs, given by the random variable X in 6.1. A patient's pneumonia status, given by the random variable Y, will lead to a variety of anatomically-relevant features A, such as increased radiopacity or consolidation in the lung fields, that form part of the

[1] Code to reproduce this project is available at `https://github.com/suinleelab/cxr\_adv`

radiograph. Furthermore, the patient's disease status will lead to a variety of clinical signs and symptoms which will influence which department they are seen in (e.g. in-patient or out-patient). Different departments may use different scanners (portable or fixed) and these scanners may be taken with different views (V). Frontal chest radiographs may be taken with either an anterior-posterior (AP) view where the x-ray source is positioned such that x-rays enter through the front of the chest and exit through the back of the chest, or a posterior-anterior (PA) view where the x-ray source is positioned such that x-rays enter through the back of the chest and exit through the front.

View directly impacts the appearance of chest radiographs in a variety of ways. Different views cause anatomical structures to have different relative sizes in radiographs since their distance from the radiographic source is altered [221]. Furthermore, AP radiographs are taken on portable scanners, which may place text such as "PORTABLE" or "SEMI-UPRIGHT" directly on the image. For this graph, it is plausible that the relationship between pneumonia and view may not be consistent across hospitals. The AP view position is generally associated with a higher prevalence of disease, as sicker patients are more likely to need to have a portable scanner brought to them [221, 304]. In our source training dataset (described below in 6.3.1), however, the standard relationship is reversed and the prevalence of pneumonia is 2-fold *higher* in PA view radiographs (2.1% base rate of pneumonia in AP images vs. 3.9% base rate of pneumonia in PA images). As the difference in base rate between the subgroups increases, the worse the generalization performance should be (see Section 6.6). Since the relationships between pneumonia and view may not be consistent across hospitals, we hypothesize that by learning a model that is invariant to differences in radiograph view, we can create a model that will be more robust to dataset shift. View is additionally an important confounder to control because commerically-available chest radiograph algorithms are currently designed to accept *both* AP and PA view radiographs as input [106].

We formally state the problem as follows. We are given data from a source distribution $\mathcal{S}$ where each sample (indexed by i) is a 3-tuple consisting of a radiograph $x_i \sim X$, a multi-label classification label $y_i \sim Y$, and a binary indicator of view $v_i \sim V$. We would like to learn a model that outputs a pneumonia score that will generalize well to a target domain $\mathcal{T}$, where the relationship between the nuisance variable and the outcome may be different in the target domain than in the source domain. In our particular problem, we assume that we have no access at all to data from the target distribution, corresponding to what Subbaswamy et al. [269] refer to as a proactive approach to addressing dataset shift. Much of the prior work on adversarial domain adaptation has corresponded to a different problem, in which we assume that we have access to *unlabeled data* from the target distribution, corresponding to what Subbaswamy et al. [269] refer to as a reactive approach to dataset shift [16, 77, 186, 115]. Since we have no data from the target distribution, we instead aim to learn a classifier f that outputs a pneumonia score S such that $S \perp V$. Even though we use all 13 of the

different pathologies in Y to train our model, since we only require that our model learns a relationship such that $S \perp V$, and not $Y \perp V$, there is no constraint for the model to learn view-independent scores for any of the other non-pneumonia pathologies.

6.3 Methods

6.3.1 Data

To assess the robustness of models to dataset shift, we used chest radiographs from two large publicly-available datasets. For our model training source domain, we used the CheXpert dataset from Stanford [109]. This dataset contains 224,316 chest radiographs of 65,240 patients. We considered only the 191,229 frontal radiographs (AP or PA view) in the dataset, excluding all of the lateral radiographs. Since the test split in the original CheXpert dataset only contained 8 radiographs that were positive for pneumonia, all of which were AP radiographs, we moved 92 more positive pneumonia radiographs (for a total of 100 positive pneumonia radiographs) to the test set for the sake of better pneumonia performance evaluation. For our target domain, we used the MIMIC-CXR dataset from Massachusetts Institute of Technology [117]. This dataset includes 371,920 chest radiographs of 65,079 patients. After filtering lateral radiographs, we had 249,995 frontal radiographs remaining. One major advantage of using these two datasets is that they have the same set of 13 labels ("Enlarged Cardiomediastinum," "Cardiomegaly," "Lung Opacity," "Lung Lesion," "Edema," "Consolidation," "Pneumonia," "Atelectasis," "Pneumothorax," "Pleural Effusion," "Pleural Other," "Fracture," and "Support Devices") and are created using the same labeling algorithm. This algorithm takes expert-generated free-text radiological reports associated with each chest radiograph as input and outputs the set of pathology labels. Using data labeled with the same natural language processing algorithm helps to remove the potential effects of dataset shift due to differences in the label generating process.

6.3.2 Standard Training

To train our baseline models for prediction, we used the architecture and training procedure described in [305] and [225]. The model architecture used was a DenseNet-121 initialized with weights pretrained on ImageNet, which can be downloaded from the PyTorch torchvision models subpackage [103, 211]. While we were primarily interested in pneumonia detection, we found that using all pathology labels available in the CheXpert dataset during training significantly increased pneumonia classification performance. Since the number of classes in the CheXpert dataset is different than the number of classes in the ImageNet dataset, the classification head for the pretrained DenseNet-121 was removed and replaced by a linear layer with output dimensions equal to the number of labels in the CheXpert dataset, followed

by a sigmoid activation function. A binary cross-entropy loss was optimized using an SGD optimizer with momentum of 0.9, weight decay of 10^{-4}, and an initial learning rate of 10^{-2}. Early stopping was implemented by monitoring binary cross-entropy loss on a held out split of validation data. Our validation set, representing 5% of the training data, was split on patients rather than radiograph index. If validation loss did not improve over an epoch, the learning rate was decreased by a factor of 10. If validation loss failed to improve for 3 consecutive epochs, training was stopped. Performance was then evaluated on the held out test set. This procedure was repeated three separate times to attain standard deviations of performance.

6.3.3 Adversarial Deconfounding

To learn more robust models that generalize better to external test data, we propose an approach based on adversarial training. This approach consists of jointly training two neural networks. The first is the classifier, f, which is trained to predict a pneumonia label y from a chest radiograph x. The second is an adversary, g, which is trained to predict the view v from the output score s of the classifier f. The optimization procedure consists of alternating between training the adversary network until it is optimal, then training the classifier to fool the adversary while still predicting pneumonia well.

This approach aims to proactively mitigate the potential effects of domain shift by controlling for known confounders in medical images using adversarial training. In addition to the applications for reactive domain adaptation mentioned above in 6.2, adversarial training has been used in a variety of other areas to learn models or representations that are independent of a given variable. For example, there is a significant body of literature in the area of algorithmic fairness where adversarial training has been used to learn representations that are fair with respect to protected classes such as race or gender [61, 183, 290]. In the physical sciences, adversarial training has been used to learn classifiers capable of detecting interesting particle jets in particle colliders that are independent of the presence of nuisance interactions in the collider [174].

We emphasize that one major contribution of our work compared to prior work on deep learning for medical images is that we take advantage of causal domain knowledge to improve generalization performance without needing to use *any* data from the target domain. Where previous approaches to domain adaptation use adversarial training to either learn a score or intermediate representation that are *domain-invariant* by augmenting training with unlabeled data from the target domain, we instead use our domain knowledge about the causal relationships involved in our data to find nuisance variables that potentially will have a different relationship with the outcome in the target domain than in the source domain. We then use an adversarial approach to learn a classifier that is invariant to the nuisance

variable, which requires no data whatsoever from the target domain.

To implement our training, we take the approach suggested in Louppe et al. [174] and adapt it for use in the application of radiograph classification. For the notation in the following sections, the parameterization of classifier f will be given as θ_f, while the parameterization of adversary g will be given by θ_g. The classifier's output score for pneumonia is given by $s = f(x)^{pneumo}$ (where the $(pneumo)$ superscript indicates the index for pneumonia in the multi-label output vector).

Separately Pretraining Classifier and Adversary

The classifier f is first trained using the procedure described in the standard training section above to optimize the negative log-likelihood of $Y|X$ under θ_f:

$$\mathcal{L}_f(\theta_f) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim Y|x}[-\log p_{\theta_f}(y|x)]. \tag{6.1}$$

Then, the parameters of the classifier are fixed and the adversary network is trained. The architecture used for the adversary is a simple feed-forward network with 3 hidden layers of 32 nodes. We used ReLU activation functions between the hidden layers, and a linear output. This architecture was selected to have sufficient capacity to model non-linear dependency between the score and view while still being lightweight enough for quick optimization. The network is optimized to minimize the following objective:

$$\mathcal{L}_r(\theta_f, \theta_r) = \mathbb{E}_{s\sim f(X;\theta_f)}\mathbb{E}_{v\sim V|s}[-\log p_{\theta_r}(v|s)]. \tag{6.2}$$

This means that the adversary takes the scalar-valued pneumonia score output by the classifier as its input, and outputs a scalar-valued prediction of view. The adversary was pretrained for a single epoch.

Joint adversarial optimization

After both the classifier and the adversary were pretrained, we began joint adversarial optimization. Each "joint optimization epoch" consisted of first fixing the classifier, then training the adversary for one epoch by minimizing the loss of the batch stochastic gradients for each of $K = N/M$ minibatches present in the entire dataset (where N is the number of total samples in the training data and M is the size of the minibatch):

$$\nabla_{\theta_r} \sum_{k=1}^{K} \sum_{m=1}^{M_k} -\log p_{\theta_r}(v_m|s_m). \tag{6.3}$$

Then, after the adversary is trained to optimally predict the nuisance variable V from the score output by the classifier, the parameters of the adversarial network θ_r are fixed, and we

draw a single minibatch of data and update the model by descending the stochastic gradients of the minibatch

$$\nabla_{\theta_f} \sum_{m=1}^{M} \big[-\log p_{\theta_f}(y_m|x_m) + \log p_{\theta_r}(v_m|s_m) \big]. \tag{6.4}$$

The procedure of an entire epoch of training for the adversary with the classifier fixed, and a single minibatch of training for the classifier with the adversary fixed, is repeated until the model achieves optimal performance while its output is independent of the nuisance variable.

Louppe et al. [174] showed that the optimal solution of this minimax optimization scheme is a classifier f that is optimal with respect to the training data with output S that is independent of V. If no such classifier exists, then the weight of the adversarial loss term given in 6.2 can be tuned with an additional hyperparameter λ to make a tradeoff between stability (in terms of independence of the classifier from the nuisance variable) and accuracy (in terms of classification performance given the data). For all of our models, we used a value of $\lambda = 1$. Finally, while other approaches have enforced independence between V and some intermediate layer of the network, if we want a pneumonia score S that is independent of V, we observe that it suffices to directly adversarially optimize the prediction of V from S.

6.3.4 Previous approaches for controlling confounders

Attempting to control for confounding in machine learning models is a well studied problem, and has previously been specifically studied in the domain of medical imaging [227]. In addition to testing the performance of our adversarial approach, we also compared to a variety of previously used approaches for modeling medical images in the presence of confounders.

Instance sampling

One approach to domain adaptation involves re-weighting samples in the training data [212, 165, 255, 271]. We re-implement the approach suggested in Rao et al., called Instance Weighting [227]. In a normal empirical risk minimization framework, we assume that the data the model will be evaluated on will be drawn from the same data generating process as that which the model is trained on, and thus aim to minimize the empirical risk:

$$f^* = \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \frac{1}{n} \ell(f(X_i), Y_i). \tag{6.5}$$

If we assume that we will have test data drawn from a different distribution, we can try to reweight the samples in our training set to minimize the empirical risk in the *target population* instead of the source population:

$$f^* = \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \frac{1}{n} \left[\frac{\hat{P}^{\mathcal{T}}(V_i, Y_i)}{\hat{P}^{\mathcal{S}}(V_i, Y_i)} \right] \ell(f(X_i), y_i), \tag{6.6}$$

where $\hat{P}^{\mathcal{S}}(V_i, Y_i)$ and $\hat{P}^{\mathcal{T}}(V_i, Y_i)$ indicate the joint density of radiograph view and pneumonia in the source and target domains respectively.

Since we do not have any information about the target distribution, we assume the target marginal distributions of the targets and the confounders are identical to the source marginal distributions, which means that the loss function factorizes to the following form:

$$f^* = \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \frac{1}{n} \left[\frac{\hat{P}^{\mathcal{S}}(Y_i)}{\hat{P}^{\mathcal{S}}(Y_i|V_i)} \right] \ell(f(X_i), y_i). \tag{6.7}$$

In order to avoid particular unbalanced batches during optimization, rather than applying the weights as a multiplicative factor during the calculation of the loss function, we instead re-weight the probability of each particular instance in the training data being sampled at each batch.

Matching

In addition to changing the sampling weights of each sample in the training set, the most straight-forward possible approach to handling confounding suggested in [227] is matching the base rate across subgroups in the training data. The drawbacks to this approach are that it either requires deliberately collecting data that is balanced across subgroups in advance, or throwing out data. Since we could not go back and alter the data collection process for our dataset, in order to match the base rate of pneumonia in AP and PA radiographs in the training data, we had to delete 77,117 AP radiographs from the training data. This represented a substantial portion of the total data, amounting to 40% of the samples negative for pneumonia in the CheXpert dataset, and 35% of all samples in the training data.

Include Nuisance Covariate in Regression

Another potential approach to handle confounding suggested in [227] is to "regress out" the effect of view on the outcome. We make use of the fact that the classification head of the DenseNet-121 is a logistic regression with the learned features (nodes of the last hidden layer, H^{n-1}) as covariates. Therefore, we simply append an extra feature for our covariate V to the last layer $H^{appended} = [H_0^{n-1}, H_1^{n-1}, \cdots, H_i^{n-1}, V]$. We can then model the data using a standard logistic regression:

$$Y = \sigma(H^{appended} w + \beta), \tag{6.8}$$

Table 6.1: Pneumonia classifier performance (AUROC ± st. dev.) on held out test data from CheXpert dataset (Source), and held out test data from the external MIMIC dataset (Target). Standard deviations reported across three independent re-initializations of the training procedure. Best performance on external test data highlighted in bold and red.

Method	Source (Internal)	Target (External)
Standard	0.791 ± 0.016	0.703 ± 0.016
Adversarial (Ours)	0.747 ± 0.013	**0.739 ± 0.001**
Instance Weighting	0.685 ± 0.049	0.648 ± 0.038
Covariate	0.793 ± 0.008	0.715 ± 0.016
Matching	0.684 ± 0.036	0.689 ± 0.024

where $w \in \mathbb{R}^{h+|V|}$ is the vector of weights of the classification head, $\beta \in \mathbb{R}$ is the bias term for the classification head, and $\sigma(t) = \frac{1}{1+e^{-t}}$.

We then train the modified DenseNet-121 following the exact same procedure as described in 6.3.2. When evaluating our model in the external target domain, we remove the effect of the confounding variable by setting it equal to the mean across all samples.

6.4 Results

6.4.1 CNN pneumonia classifiers fail to generalize to external health datasets

To assess the generalization performance of standard deep learning approaches to pneumonia classification, we trained a classifier using the procedure described in 6.3.2 on data from the Stanford CheXpert dataset, then evaluated the model on both held-out patients from the same dataset (source performance) and held-out patients from the external MIMIC dataset (target performance). We evaluated performance using area under the ROC curve (AUROC), which evaluates the true positive rate and false positive rate attainable by the model across all possible thresholds.

We found that this model was able to achieve an AUROC of pneumonia classification of 0.791 ± 0.016 (see Table 6.1). When we tested this same model on data from the PhysioNet MIMIC dataset, we found a substantial drop in performance, with the model only able to achieve an AUROC for pneumonia classification of 0.703 ± 0.016 (see Table 6.1). This result again confirms the concerns raised in [219] and [305], that state-of-the-art training and model architectures for deep learning medical imaging classifiers lead to models that do not generalize well to external datasets.

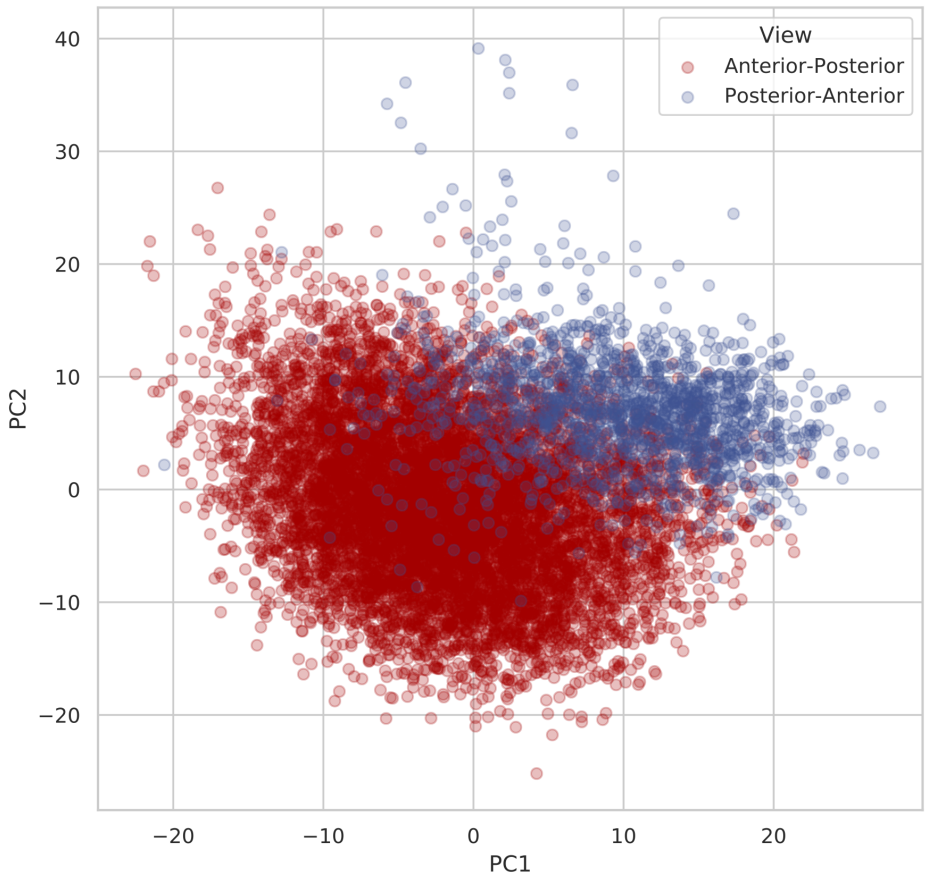

Figure 6.2: A pretrained CNN with no task-specific supervision represents radiographs in a manner easily separable by view.

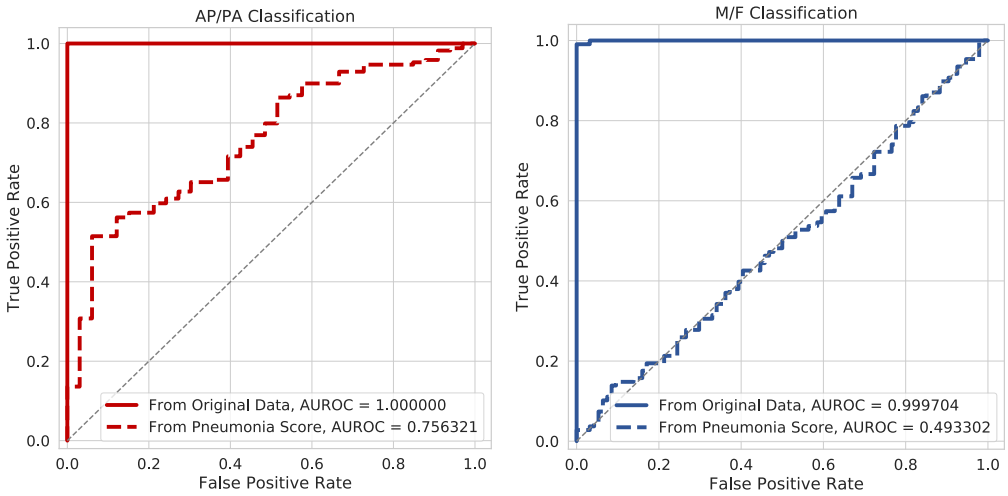

Figure 6.3: The DenseNet-121 model architecture (solid lines) is capable of near-perfect prediction of the potential nuisance variables radiograph view and patient sex from the original image data. After training a DenseNet-121 to predict pneumonia from the original image data, we see that a simple feed-forward classifier is capable of predicting radiograph view using only the scalar-valued score output by the pneumonia model as input (dashed line, left). However, a neural network classifier fails to attain better than random performance at predicting patient sex from the same scalar-valued score (dashed line, right). This indicates that the pneumonia classification score is independent of patient sex, but not of radiograph view.

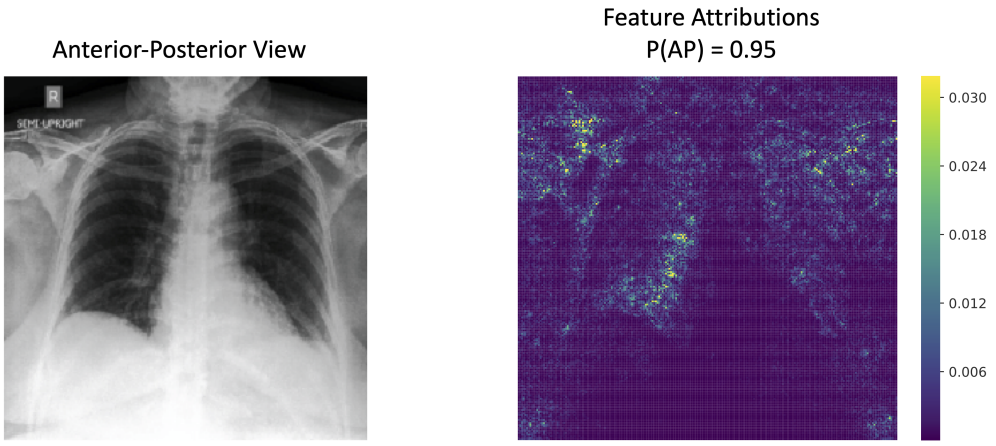

Figure 6.4: Expected Gradients feature attributions for a DenseNet-121 classifier trained to predict radiograph view position (AP vs. PA). We see that while parts of the image like the laterality markers are important (and have previously been shown to be important confounders for identifying source hospital from chest radiographs [305]), the most important pixels for identifying confounders are spread throughout the entire image, *including* within the lung fields.

6.4.2 Adversarial predictions improve model interpretability by identifying potentially confounded models

In this section we first show that state-of-the-art convolutional neural network (CNN) architectures are capable of detecting potential confounders given only pixel-level data. We then show how current approaches to model interpretability are of limited usefulness in determining whether or not a particular trained model depends on a potential confounder. We finally propose an approach based on training a neural network to predict the confounder from the model output, and show that it does a better job of identifying potential confounding.

Pretrained networks separate radiographs on basis of view and sex without any supervision

To assess how easily CNNs separate radiographs on the basis of features other than pathology, we examined the features extracted by a DenseNet-121 pretrained on ImageNet before *any* training on chest radiographs (6.2). We randomly sampled 10,000 radiographs and applied

the DenseNet-121 features submodule to them (i.e. the entire model except the classification head). We then average pooled over the last two dimensions to get 1024 features for each sample. To visualize how different sorts of radiographs were spread over these pretrained features, we performed principal components analysis on the resulting matrix, and compared the distributions of different subsets of the data along the principal components. We found that the ImageNet-pretrained DenseNet-121 easily separates chest radiographs on the basis of their view, as AP and PA radiographs are embedded in different parts of the last layer.

Pretraining may alleviate some of the effects of confounding.

Since pretraining alone was so easily able to separate confounders, we wanted to evaluate what sort of impact pretraining on ImageNet had in terms of the extent of the confounding and generalizability we observed in 6.4.1, especially in light of recent results on the potentially limited benefit of transfer learning for medical imaging [224]. Therefore, we also tried training a deep CNN architecture from randomly initialized weights using the same training approach and same CheXpert data (see Section 6.7). While we found that while this model was able to achieve comparable classification performance on held-out test data from the CheXpert dataset, it generalized significantly worse to the external target domain MIMIC data. We therefore are able to conclude that the network trained from scratch on medical data seems to learn domain-specific confounders more effectively when compared to an ImageNet pretrained model. This is somewhat intuitive, as the scratch-trained model can more easily adapt to the space of input data, including potential confounders.

CNNs can detect potential confounders from image data with high accuracy

A previously proposed approach for detecting potential confounders has been to evaluate how well that confounder can be predicted from the original data [305]. When we train the same architecture CNN using the same training procedure to predict nuisance variables like sex or radiographic view from the chest radiograph data, we find that our models are capable of predicting these variables with incredibly high accuracy (6.3). For example, we see that a model can predict view with an AUROC of 1.0, perfectly classifying every example from the held out test data. Similarly, we see that this same model architecture is capable of predicting patient sex with an AUROC of 0.9997, again on held out test data. This result establishes that even if potentially confounding nuisance variables like radiograph view or patient sex are not explicitly included in the input features of CNN classifiers, deep CNN architectures are able to extract them with high accuracy from the pixel-level features of the radiographs, allowing them to still be used in classification.

While this result indicates that CNNs can detect potential confounders from just the radiograph data, it does give us any way to tell whether or not a particular model is invariant

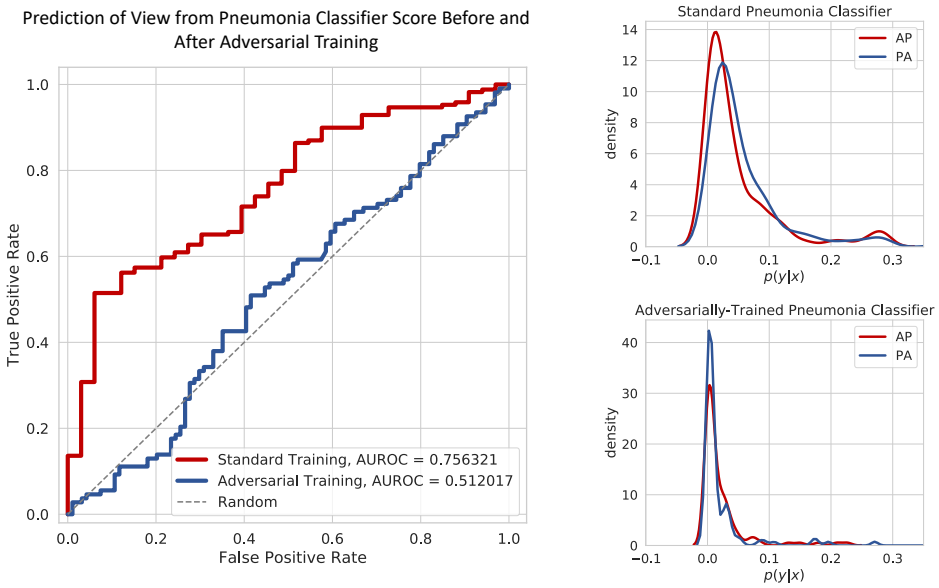

Figure 6.5: Adversarial training learns a pneumonia score that is independent of view. LEFT: ROC curves for the prediction of radiograph view (AP vs. PA) from a classifier's pneumonia score for a classifier trained with a standard approach (red) and for a classifier trained with our adversarial approach (blue). View can be predicted with relatively high accuracy just using the pneumonia score from the standard classifier, indicating that this model's output and view are not independent. After adversarial training, view can no longer be predicted with better-than-random accuracy, indicating that the output of this classifier is independent of view. RIGHT: When we look at the distribution of pneumonia scores actually output by the two models (Top: Standard, Bottom: Adversarial), we see that the distributions are not identical between AP and PA subgroups in the standard training model, but are much more closely matched betweeen the AP and PA subgroups in the adversarially-trained model.

to a particular confounder. For example, in the CheXpert dataset, the base rate of pneumonia in male patients is 2.39% while the base rate of pneumonia in female patients is 2.42%. Therefore, even though we have seen that a CNN *can* identify whether a radiograph is from a male or female patient with high accuracy, it seems likely that a model would already be invariant to a feature that does not have an association with the outcome of interest.

Saliency maps are another previously proposed approach for understanding model behavior [274, 262, 250]. These methods highlight the pixels or regions that were most important for the classifier in a given image. We therefore used Expected Gradients, a pixel-level feature attribution method [65], to generate saliency maps to help understand which pixels were important for classifying view from radiographs (see 6.4 for more details). We observe that there is no specific region in the image that is indicative of PA vs. AP view. While both the laterality marker and text marker on the image are important for classification of view, pixels throughout the entire image, including within the lung fields, are also important for this prediction. Therefore, saliency map-based approaches are also not necessarily useful for identifying whether a model is invariant to a confounder or not.

Confounders can be detected directly from score

While seeing if nuisance variables can be predicted from the images can help understand if a confounded model *could* be learned from some data, it does not help identify how much a *particular model* is actually invariant to confounders. To assess this, we instead evaluate how well a neural network model (adversary) can classify the confounder of interest using only the scalar output score of the model we care about as input. In our case, this quantifies the dependence between the output score for pneumonia S and the confounding variable V by measuring the difference between the two distributions $p(S|X, V = \text{AP})$ and $p(S|X, V = \text{PA})$, and is well-justified as an empirical approximation to the $\mathcal{H}$-divergence discussed in [77, 61]. For a classifier where the output with respect to our class of interest, S is independent of V, prediction of V from S should be random, while S not independent of V will lead to better than random prediction. As an adversary, we trained a simple feed-forward network with 3 hidden layers of 32 nodes.

While both view and sex were nearly perfectly classified from the original data, when we first train a DenseNet-121 classifier to predict pneumonia from chest radiographs, then try to predict view and sex from the predicted probability of pneumonia, we see that our model attains far greater performance at predicting radiograph view than sex, and that patient sex is not predicted better than random (6.3). Therefore, we can conclude that while the pneumonia classifier is likely independent of sex, it is potentially not invariant to view.

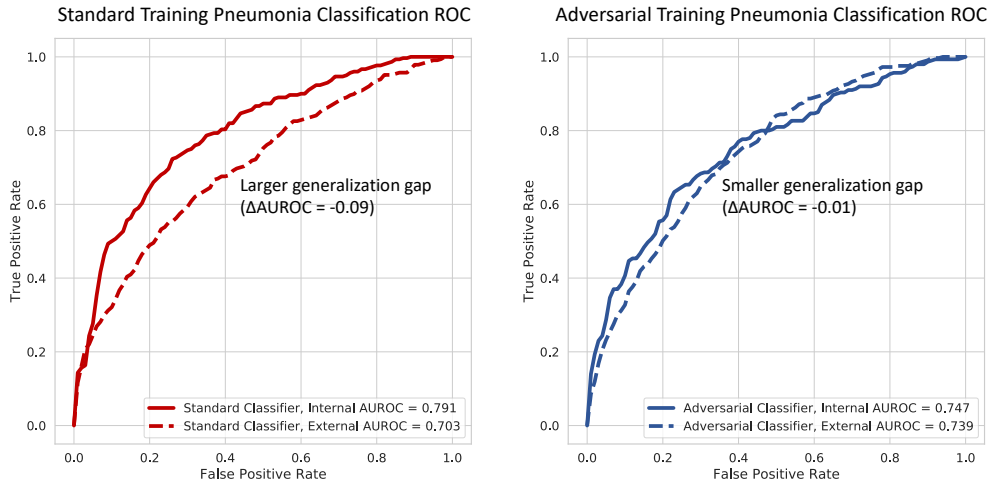

Figure 6.6: Adversarial training leads to less performance drop and significantly better performance when classifier is tested on data from a hospital system external to the one the training data comes from.

6.4.3 Adversarial training can increase model robustness by controlling for confounders

Adversarial framework learns view-independent classifier

Following the insight of the previous section, we can directly optimize for a classifier that learns a score for our class of interest S that is independent of view using an adversarial framework. Prior to adversarial training, an adversary neural network could predict the counfounder with relatively high accuracy given only the score (6.3). Following our adversarial optimization procedure (6.3.3), a neural network is not able to predict the confounder any better than random accuracy (see 6.5, left). Furthermore, when we look at the actual score distributions output by our model, we find that they are more closely matched within the two different view subgroups (see 6.5, right top and right bottom). While we mainly present results for the binary view variable, one strength of our approach is that it can be applied to any sort of nuisance variable, including continuous-valued variables like age (see Section 6.10).

We also find that looking at the predictive performance of the adversarial classifier is far more indicative of model behavior than saliency map-based approaches in this case. When we plot saliency maps (see Fig. 6.8, as described in Section 6.8) we can see that there are definite differences in the pixel-level attributions. Furthermore, it appears that the important

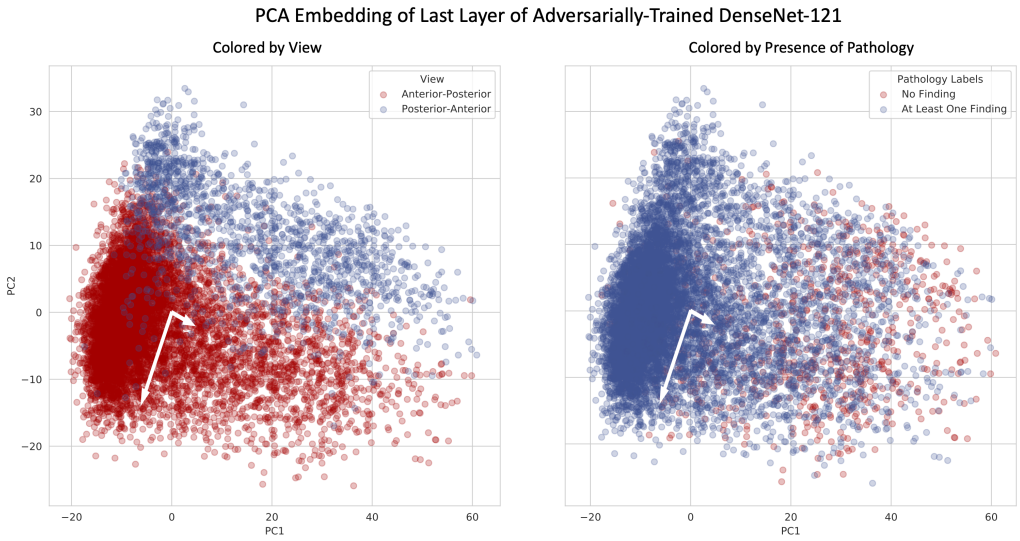

Figure 6.7: Adversarial training leads to a final representation where general pathology is orthogonal to view. White arrows indicate magnitude and direction of view and pathology classification weight vectors.

pixels are more localized to the lung fields in the adversarially-trained model than in the standard model. However, it is difficult to quantitatively assess to what extent that is the case, and since we have shown that pixels throughout the entire image are important for view classification by CNNs, it is very difficult to answer whether or not a model is confounded by view or not based only on its pixel-level feature attributions.

View-independent classifier generalizes better to unseen target domain

In addition to being able to learn a classifier that is independent of view, we find that adversarial training also is able to learn a model that generalizes better to external target domain test data (see 6.1). When we compare the performance of the adversarial model to the standard model, we find that while the adversarial model attains slightly worse performance on the source domain (AUROC $= 0.747 \pm 0.013$ vs. AUROC $= 0.791 \pm 0.016$), it attains better performance on the target domain (AUROC $= 0.739 \pm 0.001$ vs. AUROC $= 0.703 \pm 0.016$).

When we compare to the other baseline methods for controlling for confounding (instance weighting, including the covariate, and matching), we find that adversarial training also outperforms these methods. Of these other methods, we find that including the confounder as an additional covariate in modeling is the most effective, followed by matching, then the instance weighting resampling scheme.

Adversarial training learns a representation where pathology is independent of view

While our adversarial approach only explicitly constrains the final output score to be independent of the nuisance variable V representing view, we wanted to see how the earlier representations in the DenseNet-121 were impacted by this approach. We therefore take the output of the last dense layer before the classification head and average pool over the last two dimensions, and then perform principal components analysis in the same way as we did for the "unsupervised" ImageNet-pretrained classifier in section 6.4.2.

The representation in the last layer of our adversarially-trained classifier is interesting, in that it is able to learn an embedding where the axis of differentiation separating the two views seems to be orthogonal to the axis representing pathology (see 6.7). When we plot the first two principal components of the radiograph embedding and color by view, we see that the views are separated from the bottom left of the plot towards the top right. When we color the same embedding by pathology, the images with no findings are separated to the bottom right, while images containing pathology separate to the top left.

To quantify if this adversarially-trained representation has a more orthogonal relationship between view and pathology than a classifier with standard training, we learned a simple logisitic regression classifier using the first two principal components of the last layer embeddings of the standard and adversarially trained classifiers as input. The output for prediction was either view or pathology. We then measured the linear correlation (r) between the weight vectors of the two linear classifiers.

We found over a 10-fold decrease in the correlation between the view and pathology vectors in the final embeddings from the standard to adversarially-trained models (decreasing from $r = 0.1974$ to $r = 0.008$), indicating that the view-axis was substantially more orthogonal to the pathology-axis in the adversarially-trained classifier. Again, this was particularly remarkable in that there was no constraint to learn a more independent representation in the hidden layers of the model.

6.5 Discussion

Our results demonstrate that an approach based on adversarial optimization is capable of learning more robust medical imaging classifiers. For the specific case of chest radiographs, we show that a pneumonia classifier trained to be independent of view is more stable to dataset shift, attaining better generalization performance when tested on radiographs from an external dataset. Finally, our results show that attempting to predict potential nuisance variables directly from a model's output score can be a valuable tool for model interpretability, indicating whether or not a particular model is independent of potential confounders. While any measure of the difference in the distributions of the model's output conditional on

potential confounders is likely to work well, we believe that our approach is well-suited in that it also lends itself naturally to a technique to create confounder-invariant models.

Examination of the causal diagram relating chest radiographs to pneumonia points to important future research directions. Our experiments showed increased stability to dataset shift at the expense of decreased performance on new samples from the same hospital system as the training data. Given the causal diagram, where view mediates the relationship between the presence of pneumonia and the pixel features of the chest radiograph, it is not surprising that controlling for view should decrease performance. We note, however, that pneumonia is a diagnosis that is made in the context of clinical evidence of disease, and a disease where there is not necessarily perfect concordance between severity of symptoms and radiographic evidence of infiltrate [287, 201, 204, 23]. In the description of the creation of the "Pneumonia" label in the CheXpert dataset, the authors note that while pneumonia is a clinical diagnosis, "Pneumonia... was included as a label in order to represent the images that suggested primary infection as the diagnosis," suggesting that clinical information may play a role in labeling [109]. Disentangling the relationship between radiographic evidence of consolidation, the clinical presence of pneumonia symptoms, and the influence of the latter on the labeling of the former in these datasets could be helpful.

Finally, while we showed results from controlling radiograph view (and patient age), we expect that future work could show even more benefits from applying our approach to a wider variety of variables, both individually and in combination. However, it would be required for these variables to be recorded as metadata in datasets. As more and more additional variables are recorded in medical imaging datasets, and the causal relationships between these variables are better explicated, we expect the potential benefit of our approach to further increase.

6.6 Supplement: Subgroup base rate imbalance and generalization performance

Since radiograph view labels were not provided in the MIMIC dataset, we wanted to ensure that the difference in the base rate of pneumonia between view subgroups in the CheXpert training data was really an important factor contributing to poor generalization performance. Inspired by the "engineered relative risk experiment" in [305], we therefore created four synthetic subsamples of the CheXpert dataset with differing base rates of pneumonia between AP and PA radiographs. The first had a balanced ratio, the second had a 2-to-1 imbalanced ratio, the third had a 10-to-1 imabalanced ratio, and the fourth had a 100-to-1 imbalanced ratio . Each dataset contained 20,000 images total (10,000 from each AP and PA), and the base rate of pneumonia *overall* was held constant at 5%. This means that, for example, in the 100-to-1 imbalanced setting there were 10 AP radiographs that were positive for pneumonia, 9,990 AP radiographs that were negative for pneumonia, 990 PA radiographs that were

Table 6.2: Comparing extent of base rate imbalance between AP and PA in training data and generalization performance of pneumonia classification on external MIMIC test data

Base Rate Imbalance	MIMIC Pneumonia AUROC
1-to-1	0.6508
2-to-1	0.6237
10-to-1	0.5847
100-to-1	0.5749

positive for pneumonia, and 9,010 PA radiographs that were negative for pneumonia. It was necessary to decrease the sample number in the synthetic datasets to be able to achieve greater ratios of base rate imbalance between AP and PA subgroups.

We trained models on the synthetic datasets using the standard training procedure, tested on the external MIMIC dataset, then measured how much of the predictive performance was lost as the base rate became more imbalanced. In the balanced synthetic dataset, we had a baseline AUROC of 0.6508 (see 6.2. As we increased the base rate difference to 2-to-1, we saw a significant drop in predictive performance (AUROC of 0.6237). As we continued to increase the base rate difference to 10-to-1 and 100-to-1 we found that the predictive performance continued to drop. We therefore were able to conclude that as the base rate difference between AP and PA radiographs was exacerbated, the generalization performance of our classifiers was decreased.

6.7 Supplement: Randomly initialized network

To rule out the possibility that the reason these models were using confounding information and generalizing poorly was due to the initialization with weights that were pretrained on ImageNet, we also tried training deep CNNs with weights that had been randomly initialized. In addition to a DenseNet-121, we also tried training a ResNet-50 architecture [99]. Other than the change in initialization, the training procedure was identical to that described in the Standard Training section for both architectures.

While the source domain performance (area under ROC on CheXpert data) for the ResNet was comparably high to the performance attained by the ImageNet-pretrained DenseNet-121, the performance gap when testing on external target domain test data (area under ROC on MIMIC data) was actually much more significant, representing a drop in AUROC of

Table 6.3: Randomly Initialized Classifier Performance (AUROC) on Source Domain (CheXpert) and Target Domain (MIMIC)

Method	Source (Internal)	Target (External)	Δ
ResNet-50	0.7829	0.5992	-0.18
DenseNet-121	0.7098	0.5674	-0.14

0.18 (see 6.3). Furthermore, for the DenseNet architecture, both source and target domain performance decreased.

6.8 Supplement: Feature attributions

To generate saliency maps, we used the recently developed state-of-the-art method for feature attributions known as Expected Gradients [65]. Expected Gradients is an extension of the Integrated Gradients feature attribution method [274]. For a model f, the *integrated gradients* value for feature i is defined as:

$$\text{IntegratedGradients}_i(x, x') = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \delta\alpha,$$

where x is the target input and x' is baseline input. In medical imaging models, it is not clear what image would serve as a reasonable baseline. Therefore, we use the *expected gradients* value for feature i, which does not need a background reference and uses the entire training data in expectation:

$$\text{ExpectedGradients}_i(x) = \mathop{\mathbb{E}}_{x' \sim D, \alpha \sim U(0,1)} \left[(x_i - x_i') \frac{\delta f(x' + \alpha \times (x - x'))}{\delta x_i} \right].$$

.

To visualize pixel-level feature importances, we plot a heatmap where the color intensity encodes the magnitude of each attribution averaged across the three image channels. We clipped attributions in magnitude at the 99.9th percentile for the visualization.

6.9 Supplement: Subgroup base rate imbalance and adversarial confounding score

In the models trained on the synthetic datasets in 6.6, we saw that as the relationship between view and pneumonia became more confounded, the models performed less well on external test data. We wanted to ensure that the predictive performance of an adversary trained to

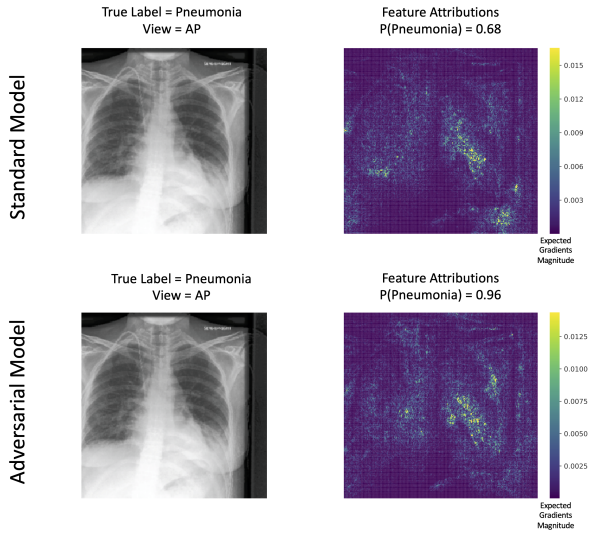

Figure 6.8: Differences in Expected Gradients attributions for standard vs. adversarial classifiers. We notice that the adversarial model places less importance on the pixels in the lower right hand corner of the image outside of the lung fields compared to the standard model. We also observe that it is difficult to quantify from pixel-level attributions how important the view confounder was to the model. This is a major limitation of saliency map approaches. While they can help rule-in the possibility of questionable model behavior, it is difficult to rule-out the possibility of undesirable confounding.

Table 6.4: Comparing extent of base rate imbalance between AP and PA in training data and performance on adversarial neural network trained to predict view from output score of classifier.

Base Rate Imbalance	Adversarial AUROC
1-to-1	0.4956
2-to-1	0.7563
10-to-1	0.9180
100-to-1	0.9436

predict view from the output of these models increased as the confounding was exacerbated. We therefore trained an adversary to predict view given the output of these three models, and found that as the confounding was exacerbated, the predictive performance of the adversary increased (see 6.4, which also includes a comparison with the adversarial performance achieved on the standard data which has a 2-to-1 imbalance).

6.10 Supplement: Age distribution results

One major advantage of the adversarial approach to deconfounding when compared to matching or balancing based approaches is that it is remarkably easy to control for continuous variables. The architecture of the adversarial neural network is changed by simply replacing the sigmoid activation function at the output with a linear activation, and replacing the binary cross-entropy loss function for the adversary network with a mean squared error loss function. Then training proceeds identically to the procedure described in 6.3.3. To demonstrate that this approach works for continuous variables, we show results for controlling the only continuous potential nusiance variable present in the CheXpert dataset, patient age.

After adversarial training a classifier to produce a pneumonia score independent of age, we wanted to evaluate how well our approach changed the age-conditional score distributions of our classifier. We split the data into four separate subgroups thresholded by age: patients younger than 45, patients with age between 45 and 65, patients between 65 and 85, and patients older than 85. We then measured all of the pairwise distances between these subgroups, in both the standard and adversarially trained classifiers. We found that in general, the score distributions were as similar or more similar between pairs of age subgroups in the adversarially-trained classifier than in the standard classifier. Since we do not expect age to be a meaningful confounder for pneumonia prediction that is likely to shift between hospital datasets, we would not expect controlling for age to improve predictive performance.

We do not find any increased predictive performance in external target domain test data by controlling for age (AUROC of 0.695 after controlling for age as compared with an AUROC of 0.703 with standard training).

To measure the difference between the age-subgroup score distributions, we used the Kolmogorov-Smirnov D statistic, which is defined as

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|, \tag{6.9}$$

where $F(x)$ is the empirical distribution function. This statistic takes values between 0 and 1, where values closer to 0 indicate more similar distributions, while values closer to 1 indicate less similar distributions. In the plot, we compare each of the six possible pairs of age subgroups on their similarity in the standard classifier and their similarity in the adversarially-trained classifier 6.9. For 3 out of the 6 pairs, we see a significant increase in score distribution similarity. For 1 pair we see a decrease, and for 2 pairs the score distribution similarity remains roughly constant. We notice that the distributions that are initially more divergent in the standard classifier are improved the most, and correspond to the greatest age difference.

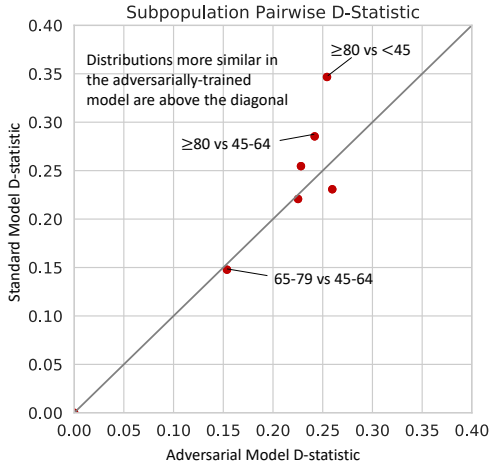

Figure 6.9: Adversarial training can learn a score that is independent of continuous random variables. To evaluate how similar the distribution of model scores were across different age subgroups, we divided the samples into four age subgroups. Each dot in the plot above represents the distance between the score distributions of two of the age subgroups (as measured by KS D-statistic). The y-axis shows the D-statistic for the scores output by the standard classification model, while the x-axis shows the D-statistic for the scores output by the adversarially-trained classification model. Dots that are above the diagonal indicate subgroups whose distributions became more similar under the adversarially-trained model.

Chapter 7

CONCLUSION

As complex machine learning models have become more prevalent in biological and medical applications, the need to understand and improve these models has grown in conjunction. The work presented in this thesis takes steps toward the development of new methods for interpreting the patterns learned by complex, black box models.

The questions left unanswered by the preceding chapters point to promising future directions for explainable AI. For example, while feature attribution methods based on Shapley values and Aumann-Shapley values come with principled and quantitative guarantees, the work in Chapter 4 demonstrated some of the shortcomings of saliency-based explanation in image data, and provided interesting preliminary results on the benefit of using of generative models to understand higher-level concepts in images. New methods capable of providing rigorous and quantitative guarantees with explanations based on concepts and features not present in the original input of models will be useful for models that take images as inputs.

Next, techniques borrowed from the statistical literature on causal inference will likely prove useful for model interpretation in both the biological sciences and medical sciences. In biological applications, researchers are often interested in inferring the importance of input features given observed data. Learning how to incorporate known information about the causal relationship between observed features and outcomes may improve the meaningfulness of the explanations learned. Furthermore, in the context of model auditing for medical models, lessons learned from this research area may help determine the causal impact of different features, shortcuts, and concepts on the deployment of models.

Finally, more human-centered studies into explainable AI methods will likely be useful. Because these methods are used for a variety of tasks, ranging from inference-like tasks to model auditing, studying which particular methods most help different users with different tasks will likely lead to further improvements.

BIBLIOGRAPHY

[1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, page 103502, 2021.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[4] Mohammed A Al-Masni, Dong-Hyun Kim, and Tae-Seong Kim. Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. *Computer Methods and Programs in Biomedicine*, 190:105351, 2020.

[5] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[6] Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

[7] Nishanth Arun, Nathan Gaw, Praveer Singh, Ken Chang, Mehak Aggarwal, Bryan Chen, Katharina Hoebel, Sharut Gupta, Jay Patel, Mishka Gidwani, et al. Assessing the (un) trustworthiness of saliency maps for localizing abnormalities in medical imaging. *medRxiv*, 2020.

[8] Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 2015.

[9] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al.

Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, 2021.

[10] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, 2021.

[11] Marcus A Badgeley, John R Zech, L. Oakden-Rayner, Benjamin S Glicksberg, Manway Liu, William Gale, Michael V McConnell, Bethany Percha, Thomas M Snyder, and Joel T Dudley. Deep learning predicts hip fracture using confounding patient and healthcare variables. *npj Digital Medicine*, 2(1):31, 2019.

[12] Noam Bar, Tal Korem, Omer Weissbrod, David Zeevi, Daphna Rothschild, Sigal Leviatan, Noa Kosower, Maya Lotan-Pompan, Adina Weinberger, Caroline I Le Roy, et al. A reference map of potential determinants for the human serum metabolome. *Nature*, 588(7836):135–140, 2020.

[13] Johnathan M Bardsley. Laplace-distributed increments, the laplace prior, and edge-preserving regularization. *J. Inverse Ill-Posed Probl*, 2012.

[14] Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

[15] Alan G Bartel, James T Chen, Robert H Peter, Victor S Behar, Yihong Kong, and Richard G Lester. The significance of coronary calcification detected by fluoroscopy: a report of 360 patients. *Circulation*, 49(6):1247–1253, 1974.

[16] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[17] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[18] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[19] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.

[20] Nicholas Bien, Pranav Rajpurkar, Robyn L Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, Bhavik N Patel, Kristen W Yeom, Katie Shpanskaya, et al. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet. *PLoS medicine*, 15(11):e1002699, 2018.

[21] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[22] Florian J Bock, Catherine Cloix, Desiree Zerbst, and Stephen WG Tait. Apoptosis-induced fgf signalling promotes non-cell autonomous resistance to cell death. *bioRxiv*, 2020.

[23] Bálint Botz. A Few Thoughts About CheXNet — And The Way Human Performance Should (And Should Not) Be Measured. *Web*, pages 1–4, 2017.

[24] B Braithwaite, J Paananen, Heidi Taipale, Antti Tanskanen, Jari Tiihonen, Sirpa Hartikainen, and Anna-Maija Tolppanen. Detection of medications associated with alzheimer's disease using ensemble methods and cooperative game theory. *International Journal of Medical Informatics*, 141:104142, 2020.

[25] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[26] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[27] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[28] Keno K Bressem, Lisa Adams, Christoph Erxleben, Bernd Hamm, Stefan Niehues, and Janis Vahldiek. Comparing different deep learning architectures for classification of chest radiographs. *arXiv:2002.08991*, 2020.

[29] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[30] Luca Brunese, Francesco Mercaldo, Alfonso Reginelli, and Antonella Santone. Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays. *Computer Methods and Programs in Biomedicine*, 196:105608, 2020.

[31] Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. *arXiv preprint*, 2019.

[32] Kevin D Bunting, Wen-Mei Yu, Heath L Bradley, Eleonora Haviernikova, Ann E Kelly-Welch, Achsah D Keegan, and Cheng-Kui Qu. Increased numbers of committed myeloid progenitors but not primitive hematopoietic stem/progenitors in mice lacking stat6 expression. *Journal of leukocyte biology*, 76(2):484–490, 2004.

[33] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797, 2020.

[34] Danilo Bzdok, Denis Engemann, and Bertrand Thirion. Inference and prediction diverge in biomedicine. *Patterns*, 1(8):100119, 2020.

[35] Diego Calzolari, Stefania Bruschi, Laurence Coquin, Jennifer Schofield, Jacob D Feala, John C Reed, Andrew D McCulloch, and Giovanni Paternostro. Search algorithms as a framework for the optimization of drug combinations. *PLoS Comput Biol*, 4(12):e1000249, 2008.

[36] Daniel C Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):1–10, 2020.

[37] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

[38] Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. In *Advances in Neural Information Processing Systems*, pages 14300–14310, 2019.

[39] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[40] Wei Cheng, Xiang Zhang, Zhishan Guo, Yu Shi, and Wei Wang. Graph-regularized dual Lasso for robust eQTL mapping. *Bioinformatics*, 30(12):i139–i148, 06 2014.

[41] Ting-Chao Chou. Drug combination studies and their synergy quantification using the chou-talalay method. *Cancer research*, 70(2):440–446, 2010.

[42] Jane E Churpek, Khateriaa Pyrtel, Krishna-Latha Kanchi, Jin Shao, Daniel Koboldt, Christopher A Miller, Dong Shen, Robert Fulton, Michelle O'Laughlin, Catrina Fronick, et al. Genomic analysis of germ line and somatic variants in familial myelodysplasia/acute myeloid leukemia. *Blood*, 126(22):2484–2490, 2015.

[43] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong, and Marzyeh Ghassemi. Covid-19 image data collection: prospective predictions are the future. *arXiv 2006.11988*, 2020.

[44] M Ryan Corces, Jason D Buenrostro, Beijing Wu, Peyton G Greenside, Steven M Chan, Julie L Koenig, Michael P Snyder, Jonathan K Pritchard, Anshul Kundaje, William J Greenleaf, et al. Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nature genetics*, 48(10):1193–1203, 2016.

[45] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *arXiv preprint arXiv:2011.14878*, 2020.

[46] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions through additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

[47] David Croft, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R Kamdar, et al. The reactome pathway knowledgebase. *Nucleic acids research*, 42(D1):D472–D477, 2014.

[48] Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Recovering pairwise interactions using neural networks. *arXiv preprint arXiv:1901.08361*, 2019.

[49] Resul Das, Ibrahim Turkoglu, and Abdulkadir Sengur. Effective diagnosis of heart disease through neural networks ensembles. *Expert systems with applications*, 36(4):7675–7680, 2009.

[50] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 598–617. IEEE, 2016.

[51] Daphne Day and Lillian L Siu. Approaches to modernize the combination drug development paradigm. *Genome medicine*, 8(1):1–14, 2016.

[52] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, pages 1–10, 2021.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[54] Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64(5):304–310, 1989.

[55] Robert Detrano, Ernesto E. Salcedo, Robert E. Hobbs, and John Yiannikas. Cardiac cinefluoroscopy as an inexpensive aid in the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 57(13):1041 – 1046, 1986.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[57] Kedar Dhamdhere, Ashish Agarwal, and Mukund Sundararajan. The shapley taylor interaction index. *arXiv preprint arXiv:1902.05622*, 2019.

[58] Ayse Berceste Dincer, Safiye Celik, Naozumi Hiranuma, and Su-In Lee. Deepprofile: Deep learning of cancer molecular profiles for precision medicine. *BioRxiv*, page 278739, 2018.

[59] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, pages 13567–13578, 2019.

[60] Jiayun Dong and Cynthia Rudin. Variable importance clouds: A way to explore variable importance for the set of good models. *arXiv preprint arXiv:1901.03209*, 2019.

[61] Harrison Edwards and Amos Storkey. Censoring Representations with an Adversary. 11 2015.

[62] Bradley Efron. Prediction, estimation, and attribution. *Journal of the American Statistical Association*, 115(530):636–655, 2020.

[63] Edith Elkind, Leslie Ann Goldberg, Paul W Goldberg, and Michael Wooldridge. On the computational complexity of weighted voting games. *Annals of Mathematics and Artificial Intelligence*, 56(2):109–131, 2009.

[64] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[65] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019.

[66] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

[67] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[68] Negar Farzaneh, Craig A Williamson, Jonathan Gryak, and Kayvan Najarian. A hierarchical expert-guided machine learning framework for clinical decision support systems: an application to traumatic brain injury prognostication. *npj Digital Medicine*, 4(1):78, 2021.

[69] Jacob D Feala, Jorge Cortes, Phillip M Duxbury, Carlo Piermarocchi, Andrew D McCulloch, and Giovanni Paternostro. Systems approaches and algorithms for discovery of combinatorial therapies. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 2(2):181–193, 2010.

[70] Jean Feng and Noah Simon. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*, 2017.

[71] Ramón Flores, Elisenda Molina, and Juan Tejada. Evaluating groups with the generalized shapley value. *4OR*, 17(2):141–172, 2019.

[72] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018.

[73] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[74] Eric J Friedman. Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32(4):501–518, 2004.

[75] H Gal, N Amariglio, L Trakhtenbrot, J Jacob-Hirsh, O Margalit, A Avigdor, A Nagler, S Tavor, L Ein-Dor, T Lapidot, et al. Gene expression profiles of aml derived stem cells; similarity to hematopoietic stem cells. *Leukemia*, 20(12):2147–2154, 2006.

[76] William Gale, L. Oakden-Rayner, Gustavo Carneiro, Andrew P Bradley, and Lyle J Palmer. Detecting hip fractures with radiologist-level performance using deep neural networks. *arXiv:1711.06504*, 2017.

[77] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2030–2096, 2016.

[78] Mathew J Garnett, Elena J Edelman, Sonja J Heidorn, Chris D Greenman, Anahita Dastur, King Wai Lau, Patricia Greninger, I Richard Thompson, Xi Luo, Jorge Soares, et al. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575, 2012.

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

[80] J Raymond Geis, Adrian P Brady, Carol C Wu, Jack Spencer, Erik Ranschaert, Jacob L Jaremko, Steve G Langer, Andrea Borondy Kitts, Judy Birch, William F Shields, et al. Ethics of artificial intelligence in radiology: summary of the joint European and North American multisociety statement. *Radiology*, 293(2):436–440, 2019.

[81] Andrew J Gentles, Sylvia K Plevritis, Ravindra Majeti, and Ash A Alizadeh. Association of a leukemic stem cell gene expression signature with clinical outcomes in acute myeloid leukemia. *Jama*, 304(24):2706–2715, 2010.

[82] Reza Ghaeini, Xiaoli Z Fern, and Prasad Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language inference. *arXiv preprint arXiv:1808.03894*, 2018.

[83] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[84] Biraja Ghoshal and Allan Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv:2003.10769*, 2020.

[85] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[86] Shreevrat Goenka and Mark H Kaplan. Transcriptional regulation by stat6. *Immunologic research*, 50(1):87–96, 2011.

[87] Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

[88] Casey S Greene, Arjun Krishnan, Aaron K Wong, Emanuela Ricciotti, Rene A Zelaya, Daniel S Himmelstein, Ran Zhang, Boris M Hartmann, Elena Zaslavsky, Stuart C Sealfon, et al. Understanding multicellular function and disease with human tissue-specific networks. *Nature genetics*, 47(6):569, 2015.

[89] Peyton Greenside, Tyler Shimko, Polly Fordyce, and Anshul Kundaje. Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics*, 34(17):i629–i637, 2018.

[90] Jinyang Gu, Bing Han, and Jian Wang. Covid-19: gastrointestinal manifestations and potential fecal-oral transmission. *Gastroenterology*, 158(6):1518–1519, 2020.

[91] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, and Jorge Cuadros. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[92] Lindsay M Gurska, Kristina Ames, and Kira Gritsman. Signaling pathways in leukemic stem cells. *Leukemia Stem Cells in Hematologic Malignancies*, pages 1–39, 2019.

[93] Gilles Gut, Stefan G Stark, Gunnar Rätsch, and Natalie R Davidson. Pmvae: Learning interpretable single-cell representations with pathway modules. *bioRxiv*, 2021.

[94] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[95] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[96] Lina Han, Qi Zhang, Monique Dail, Ce Shi, Antonio Cavazos, Vivian R Ruvolo, Yang Zhao, Eugene Kim, Mohamed Rahmani, Duncan H Mak, et al. Concomitant targeting of bcl2 with venetoclax and mapk signaling with cobimetinib in acute myeloid leukemia models. *Haematologica*, 105(3):697, 2020.

[97] Jie Hao, Youngsoon Kim, Tae-Kyung Kim, and Mingon Kang. Pasnet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data. *BMC bioinformatics*, 19(1):1–13, 2018.

[98] Stephanie A. Harmon, Thomas H. Sanford, Sheng Xu, Evrim B. Turkbey, Holger Roth, Ziyue Xu, Dong Yang, Andriy Myronenko, Victoria Anderson, Amel Amalou, Maxime Blain, Michael Kassin, Dilara Long, Nicole Varble, Stephanie M. Walker, Ulas Bagci, Anna Maria Ierardi, Elvira Stellato, Guido Giovanni Plensich, Guiseppe Franceschelli, Cristiano Girlando, Giovanni Irmici, Dominic Labella, Dima Hammoud, Ashkan Malayeri, Elizabeth Jones, Ronald M. Summers, Peter L. Choyke, Daguang Xu, Mona Flores, Kaku Tamura, Hirofumi Obinata, Hitoshi Mori, Francesca Patella, Maurizio Cariati, Gianpaolo Carrafiello, Peng An, Bradford J. Wood, and Baris Turkbey. Artificial intelligence for the detection of covid-19 pneumonia on chest ct using multinational datasets. *Nature Communications*, 11:4080, 2020.

[99] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[100] Ezz El-Din Hemdan, Marwa A Shouman, and Mohamed Esmail Karar. Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv:2003.11055*, 2020.

[101] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, 2019.

[102] Andrew L Hopkins. Network pharmacology: the next paradigm in drug discovery. *Nature chemical biology*, 4(11):682–690, 2008.

[103] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. 8 2016.

[104] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[105] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[106] Eui Jin Hwang, Ju Gang Nam, Woo Hyeon Lim, Sae Jin Park, Yun Soo Jeong, Ji Hee Kang, Eun Kyoung Hong, Taek Min Kim, Jin Mo Goo, Sunggyun Park, Ki Hwan Kim, and Chang Min Park. Deep Learning for Chest Radiograph Diagnosis in the Emergency Department. *Radiology*, 293(3):573–580, 10 2019.

[107] Stephanie L Hyland, Martin Faltys, Matthias Hüser, Xinrui Lyu, Thomas Gumbsch, Cristóbal Esteban, Christian Bock, Max Horn, Michael Moor, Bastian Rieck, et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature medicine*, 26(3):364–373, 2020.

[108] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[109] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, and Katie Shpanskaya. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *arXiv preprint arXiv:1901.07031*, 2019.

[110] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[111] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018.

[112] Joseph D Janizek, Ayse Berceste Dincer, Safiye Celik, Hugh Chen, William Chen, Kamila Naxerova, and Su-In Lee. Uncovering expression signatures of synergistic drug response using an ensemble of explainable ai models. *bioRxiv*, 2021.

[113] Joseph D Janizek, Gabriel Erion, Alex J DeGrave, and Su-In Lee. An adversarial approach for the robust classification of pneumonia from chest radiographs. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 69–79, 2020.

[114] Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(104):1–54, 2021.

[115] Mehran Javanmardi and Tolga Tasdizen. Domain adaptation for biomedical image segmentation using adversarial training. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 554–558. IEEE, 2018.

[116] Jia Jia, Feng Zhu, Xiaohua Ma, Zhiwei W Cao, Yixue X Li, and Yu Zong Chen. Mechanisms of drug combinations: interaction and network perspectives. *Nature reviews Drug discovery*, 8(2):111–128, 2009.

[117] Alistair E W Johnson, Tom J Pollard, Seth Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. MIMIC-CXR: A large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[118] Courtney L Jones, Brett M Stevens, Angelo D'Alessandro, Rachel Culp-Hill, Julie A Reisz, Shanshan Pei, Annika Gustafson, Nabilah Khan, James DeGregori, Daniel A Pollyea, et al. Cysteine depletion targets leukemia stem cells through inhibition of electron transport complex ii. *Blood, The Journal of the American Society of Hematology*, 134(4):389–394, 2019.

[119] Minoru Kanehisa et al. The kegg database. In *Novartis Foundation Symposium*, pages 91–100. Wiley Online Library, 2002.

[120] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Segment integrated gradients: Better attributions through regions. *arXiv preprint arXiv:1906.02825*, 2019.

[121] Md Karim, Till Döhmen, Dietrich Rebholz-Schuhmann, Stefan Decker, Michael Cochez, Oya Beyan, et al. Deepcovidexplainer: Explainable covid-19 predictions based on chest x-ray images. *arXiv:2004.04582*, 2020.

[122] Riikka Karjalainen, Minxia Liu, Ashwini Kumar, Liye He, Disha Malani, Alun Parsons, Mika Kontro, Olli Kallioniemi, Kimmo Porkka, and Caroline A Heckman. Elevated expression of s100a8 and s100a9 correlates with resistance to the bcl-2 inhibitor venetoclax in aml. *Leukemia*, 33(10):2548–2553, 2019.

[123] MJ Keith, A Jameson, W Van Straten, M Bailes, S Johnston, M Kramer, A Possenti, SD Bates, NDR Bhat, M Burgay, et al. The high time resolution universe pulsar survey–i. system configuration and initial discoveries. *Monthly Notices of the Royal Astronomical Society*, 409(2):619–627, 2010.

[124] Asim Khwaja, Magnus Bjorkholm, Rosemary E Gale, Ross L Levine, Craig T Jordan, Gerhard Ehninger, Clara D Bloomfield, Eli Estey, Alan Burnett, Jan J Cornelissen, et al. Acute myeloid leukaemia. *Nature reviews Disease primers*, 2(1):1–22, 2016.

[125] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

[126] Hui Kwon Kim, Goosang Yu, Jinman Park, Seonwoo Min, Sungtae Lee, Sungroh Yoon, and Hyongbum Henry Kim. Predicting the efficiency of prime editing guide rnas in human cells. *Nature Biotechnology*, 39(2):198–206, 2021.

[127] Nahye Kim, Hui Kwon Kim, Sungtae Lee, Jung Hwa Seo, Jae Woo Choi, Jinman Park, Seonwoo Min, Sungroh Yoon, Sung-Rae Cho, and Hyongbum Henry Kim. Prediction of the sequence-specific cleavage activity of cas9 variants. *Nature Biotechnology*, 38(11):1328–1336, 2020.

[128] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[129] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[130] Ross D King, Oghenejokpeme I Orhobor, and Charles C Taylor. Cross-validation is safe to use. *Nature Machine Intelligence*, 3(4):276–276, 2021.

[131] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[132] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR*, abs/1609.0, 2016.

[133] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

[134] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[135] Igor Kononenko et al. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18, 2010.

[136] Peter K Koo, Praveen Anand, Steffan B Paul, and Sean R Eddy. Inferring sequence-structure preferences of rna-binding proteins with convolutional residual networks. *bioRxiv*, page 418459, 2018.

[137] Peter K Koo and Matt Ploenzke. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *Nature Machine Intelligence*, 3(3):258–266, 2021.

[138] Steven M Kornblau, David McCue, Neera Singh, Wenjing Chen, Zeev Estrov, and Kevin R Coombes. Recurrent expression signatures of cytokines and chemokines are present and are independently prognostic in acute myelogenous leukemia and myelodysplasia. *Blood, The Journal of the American Society of Hematology*, 116(20):4251–4261, 2010.

[139] PM Korthuis, G Berger, B Bakker, M Rozenveld-Geugien, J Jaques, G De Haan, JJ Schuringa, E Vellenga, and H Schepers. Cited2-mediated human hematopoietic stem cell maintenance is critical for acute myeloid leukemia. *Leukemia*, 29(3):625–635, 2015.

[140] Kamil R Kranc, Hein Schepers, Neil P Rodrigues, Simon Bamforth, Ellen Villadsen, Helen Ferry, Tiphaine Bouriez-Jones, Mikael Sigvardsson, Shoumo Bhattacharya, Sten Eirik Jacobsen, et al. Cited2 is an essential regulator of adult hematopoietic stem cells. *Cell stem cell*, 5(6):659–665, 2009.

[141] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[142] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *2012 Conference on Neural Information Processing Systems*, 2012.

[143] Brent M Kuenzi, Jisoo Park, Samson H Fong, Kyle S Sanchez, John Lee, Jason F Kreisberg, Jianzhu Ma, and Trey Ideker. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer cell*, 38(5):672–684, 2020.

[144] Ashish R Kumar, Aaron L Sarver, Baolin Wu, and John H Kersey. Meis1 maintains stemness signature in mll-af9 leukemia. *Blood*, 115(17):3642–3643, 2010.

[145] Shinjini Kundu. Ai in medicine must be explainable. *Nature Medicine*, pages 1–1, 2021.

[146] Shinjini Kundu, Hesham Elhalawani, Judy W Gichoya, and Charles E Kahn Jr. How might ai and chest imaging help unravel covid-19's mysteries? *Radiology. Artificial Intelligence*, 2(3), 2020.

[147] Stephen E Kurtz, Christopher A Eide, Andy Kaempf, Vishesh Khanna, Samantha L Savage, Angela Rofelty, Isabel English, Hibery Ho, Ravi Pandya, William J Bolosky, Hoifung Poon, Michael W Deininger, Robert Collins, Ronan T Swords, Justin Watts, Daniel A Pollyea, Bruno C Medeiros, Elie Traer, Cristina E Tognon, Motomi Mori, Brian J Druker, and Jeffrey W Tyner. Molecularly targeted drug combinations demonstrate selective effectiveness for myeloid- and lymphoid-derived hematologic malignancies. *Proceedings of the National Academy of Sciences*, page 201703094, 8 2017.

[148] Stephen E Kurtz, Christopher A Eide, Andy Kaempf, Motomi Mori, Cristina E Tognon, Uma Borate, Brian J Druker, and Jeffrey W Tyner. Dual inhibition of jak1/2 kinases and bcl2: a promising therapeutic strategy for acute myeloid leukemia. *Leukemia*, 32(9):2025–2028, 2018.

[149] Heikki Kuusanmäki, Aino-Maija Leppä, Petri Pölönen, Mika Kontro, Olli Dufva, Debashish Deb, Bhagwan Yadav, Oscar Brück, Ashwini Kumar, Hele Everaus, et al. Phenotype-based drug screening reveals association between venetoclax response and differentiation stage in acute myeloid leukemia. *Haematologica*, 105(3):708, 2020.

[150] Andrea Laghi. Cautions about radiologic diagnosis of covid-19 infection driven by artificial intelligence. *The Lancet Digital Health*, 2(5):e225, 2020.

[151] Vivian Lai, Jon Z Cai, and Chenhao Tan. Many faces of feature importance: Comparing built-in and post-hoc feature importance in text classification. *arXiv preprint arXiv:1910.08534*, 2019.

[152] Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017.

[153] Jatinder K Lamba. Genetic factors influencing cytarabine therapy. *Pharmacogenomics*, 10(10):1657–1674, 2009.

[154] H Lannert, M Lenze, T Able, A Lenze, R Saffrich, V Eckstein, S Leicht, X Li, T Franz, and AD Ho. Expression of s100 proteins in normal human hematopoietic stem cells and in aml. *Journal of Clinical Oncology*, 26(15_suppl):7072–7072, 2008.

[155] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

[156] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126, 2017.

[157] Su-In Lee, Safiye Celik, Benjamin A Logsdon, Scott M Lundberg, Timothy J Martins, Vivian G Oehler, Elihu H Estey, Chris P Miller, Sylvia Chien, Jin Dai, et al. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nature communications*, 9(1):1–13, 2018.

[158] Jeffrey T Leek, W Evan Johnson, Hilary S Parker, Andrew E Jaffe, and John D Storey. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, 28(6):882–883, 2012.

[159] Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLOS Genetics*, 3(9):1–12, 09 2007.

[160] Matthew D Li, Nishanth Thumbavanam Arun, Mishka Gidwani, Ken Chang, Francis Deng, Brent P Little, Dexter P Mendoza, Min Lang, Susanna I Lee, Aileen O'Shea, et al. Automated assessment and tracking of covid-19 pulmonary disease severity on chest radiographs using convolutional siamese neural networks. *Radiology: Artificial Intelligence*, 2(4):e200079, 2020.

[161] Mulin Jun Li, Panwen Wang, Xiaorong Liu, Ee Lyn Lim, Zhangyong Wang, Meredith Yeager, Maria P Wong, Pak Chung Sham, Stephen J Chanock, and Junwen Wang. Gwasdb: a database for human genetic variants identified by genome-wide association studies. *Nucleic acids research*, 40(D1):D1047–D1054, 2012.

[162] Wenyu Lin, Chaoqun Wang, Guangping Liu, Chao Bi, Xian Wang, Qiyin Zhou, and Hongchuan Jin. SLC7A11/xCT in cancer: biological functions and therapeutic implications. *American journal of cancer research*, 10(10):3106–3126, 10 2020.

[163] Zhong Qiu Lin, Mohammad Javad Shafiee, Stanislav Bochkarev, Michael St Jules, Xiao Yu Wang, and Alexander Wong. Explaining with impact: A machine-centric strategy to quantify the performance of explainability algorithms. *arXiv preprint arXiv:1910.07387*, 2019.

[164] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[165] Max A Little and Reham Badawy. Causal bootstrapping. *arXiv preprint arXiv:1910.09648*, 2019.

[166] Frederick Liu and Besim Avci. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286*, 2019.

[167] Jiangying Liu, Ya-Zhen Qin, Shenmiao Yang, Yazhe Wang, Ying-Jun Chang, Ting Zhao, Qian Jiang, and Xiao-Jun Huang. Meis1 is critical to the maintenance of human acute myeloid leukemia cells independent of mll rearrangements. *Annals of hematology*, 96(4):567–574, 2017.

[168] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*, pages 730–734. IEEE, 2015.

[169] Wei Liu, Yue Zhang, Cheuk-Man Yu, Qing-Wei Ji, Meng Cai, Ying-Xin Zhao, and Yu-Jie Zhou. Current understanding of coronary artery calcification. *Journal of geriatric cardiology: JGC*, 12(6):668, 2015.

[170] Fang-Yu Lo, Kit Man Ng, Wen-Chun Chen, Chung-Yi Hu, Hsin-An Hou, Cheng-Hong Tsai, Da-Liang Ou, Hwei-Fang Tien, and Liang-In Lin. Metabolic alterations may contribute to cabozantinib resistance in acute myeloid leukemia cells with flt3-itd. *Blood*, 132:2785, 2018.

[171] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

[172] Yifei Lou, Tieyong Zeng, Stanley Osher, and Jack Xin. A weighted difference of anisotropic and isotropic total variation model for image processing. *SIAM Journal on Imaging Sciences*, 8(3):1798–1823, 2015.

[173] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *arXiv preprint arXiv:1705.08821*, 2017.

[174] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. In *Advances in neural information processing systems*, pages 981–990, 2017.

[175] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1–21, 2014.

[176] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. Explainable ai for trees: From local explanations to global understanding, 2019.

[177] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[178] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature: Machine Intelligence*, 2020.

[179] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[180] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[181] Robert J Lyon, BW Stappers, Sally Cooper, JM Brooke, and JD Knowles. Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1):1104–1123, 2016.

[182] Angela HEM Maas and Yolande EA Appelman. Gender differences in coronary heart disease. *Netherlands Heart Journal*, 18(12):598–603, 2010.

[183] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018.

[184] Gianluca Maguolo and Loris Nanni. A critic evaluation of methods for covid-19 automatic detection from x-ray images. *arXiv:2004.12823*, 2020.

[185] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[186] Faisal Mahmood, Richard Chen, and Nicholas J Durr. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE transactions on medical imaging*, 37(12):2572–2581, 2018.

[187] Jean-Luc Marichal, Ivan Kojadinovic, and Katsushige Fujimoto. Axiomatic characterizations of generalized values. *Discrete Applied Mathematics*, 155(1):26–43, 2007.

[188] Alexandra Maslova, Ricardo N Ramirez, Ke Ma, Hugo Schmutz, Chendi Wang, Curtis Fox, Bernard Ng, Christophe Benoist, Sara Mostafavi, et al. Deep learning of immune cell differentiation. *Proceedings of the National Academy of Sciences*, 117(41):25655–25666, 2020.

[189] Florian Meier, Niklas D Köhler, Andreas-David Brunner, Jean-Marc H Wanka, Eugenia Voytik, Maximilian T Strauss, Fabian J Theis, and Matthias Mann. Deep learning the collisional cross sections of the peptide universe from a million experimental values. *Nature communications*, 12(1):1–12, 2021.

[190] Michael P Menden, Dennis Wang, Mike J Mason, Bence Szalai, Krishna C Bulusu, Yuanfang Guan, Thomas Yu, Jaewoo Kang, Minji Jeon, Russ Wolfinger, et al. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature communications*, 10(1):1–17, 2019.

[191] Henry W Miller. Plan and operation of the health and nutrition examination survey, united states, 1971-1973. *DHEW publication no.(PHS)-Dept. of Health, Education, and Welfare (USA)*, 1973.

[192] Akinori Mitani, Abigail Huang, Subhashini Venugopalan, Greg S Corrado, Lily Peng, Dale R Webster, Naama Hammel, Yun Liu, and Avinash V Varadarajan. Detection of anaemia from retinal fundus images via deep learning. *Nature Biomedical Engineering*, 4(1):18–27, 2020.

[193] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[194] Mahmud Mossa-Basha, Jonathan Medverd, Kenneth Linnau, John B Lynch, Mark H Wener, Gregory Kicska, Thomas Staiger, and Dushyant Sahani. Policies and guidelines for covid-19 preparedness: Experiences from the university of washington. *Radiology*, page 201326, 2020.

[195] W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*, 2018.

[196] Keelin Murphy, Henk Smits, Arnoud J.G. Knoops, Michael B. J. M. Korst, Tijs Samson, Ernst T. Scholten, Steven Schalekamp, Cornelia M. Schaefer-Prokop, Rick H. H. M. Philipsen, Annet Meijers, Jaime Melendez, Bram van Ginneken, and Matthieu Rutten. Covid-19 on chest radiographs: a multireader evaluation of an artificial intelligence system. *Radiology*, 296(3):E166–E172, 2020.

[197] Nikhil Naik, Ali Madani, Andre Esteva, Nitish Shirish Keskar, Michael F Press, Daniel Ruderman, David B Agus, and Richard Socher. Deep learning-enabled breast cancer hormonal receptor status determination from base-level h&e stains. *Nature communications*, 11(1):1–8, 2020.

[198] Ramya Nair, Alejandro Salinas-Illarena, and Hanna-Mari Baldauf. New strategies to treat aml: novel insights into aml survival pathways and combination therapies. *Leukemia*, 35(2):299–311, 2021.

[199] Yadati Narahari. *Game theory and mechanism design*, volume 4. World Scientific, 2014.

[200] Ming-Yen Ng, Elaine YP Lee, Jin Yang, Fangfang Yang, Xia Li, Hongxia Wang, Macy Mei-sze Lui, Christine Shing-Yen Lo, Barry Leung, Pek-Lan Khong, Christopher Kim-Ming-Hui, Kwok-yung Yuen, and Michael David Kuo. Imaging profile of the covid-19 infection: Radiologic findings and literature review. *Radiology: Cardiothoracic Imaging*, 2(1):e200034, 2020.

[201] M S Niederman, J B Bass, G Douglas Campbell, A M Fein, R F Grossman, L A Mandell, T J Marrie, A Torres, and V L Yu. Guidelines for the initial management of adults with community-acquired pneumonia: diagnosis, assessment of severity, and initial antimicrobial therapy. *American Review of Respiratory Disease*, 148(5):1418–1426, 1993.

[202] Darryl Nishimura. Biocarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, 2(3):117–120, 2001.

[203] B. V. North, D. Curtis, and P. C. Sham. A note on the calculation of empirical p values from monte carlo procedures. *American Journal of Human Genetics*, 71(2):439–441, 2002.

[204] Lauren Oakden-Rayner. CheXNet: an in-depth review. *URL: https://laurenoakdenrayner.com/2018/01/24/chexnetan-in-depth-review/*, 2018.

[205] Radiological Society of North America. Rsna pneumonia detection challenge.

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

[207] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

[208] Jennifer O'Neil, Yair Benita, Igor Feldman, Melissa Chenard, Brian Roberts, Yaping Liu, Jing Li, Astrid Kral, Serguei Lejnine, Andrey Loboda, et al. An unbiased oncology compound screen to identify novel combination strategies. *Molecular cancer therapeutics*, 15(6):1155–1162, 2016.

[209] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine*, page 103792, 2020.

[210] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[211] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

[212] Judea Pearl and Elias Bareinboim. Transportability across studies: A formal approach. Technical report, 2011.

[213] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[214] Shanshan Pei, Daniel A Pollyea, Annika Gustafson, Brett M Stevens, Mohammad Minhajuddin, Rui Fu, Kent A Riemondy, Austin E Gillen, Ryan M Sheridan, Jihye Kim, et al. Monocytic subclones confer resistance to venetoclax-based therapy in patients with acute myeloid leukemia. *Cancer discovery*, 10(4):536–551, 2020.

[215] P Peña-Martínez, Mia Eriksson, R Ramakrishnan, M Chapellier, Carl Högberg, C Orsmark-Pietras, J Richter, Anna Andersson, T Fioretos, and M Järås. Interleukin 4 induces apoptosis of acute myeloid leukemia cells in a stat6-dependent manner. *Leukemia*, 32(3):588–596, 2018.

[216] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[217] Daniel A Pollyea, Maria Amaya, Paolo Strati, and Marina Y Konopleva. Venetoclax for aml: changing the treatment paradigm. *Blood advances*, 3(24):4326–4335, 2019.

[218] Daniel A Pollyea, Brett M Stevens, Courtney L Jones, Amanda Winters, Shanshan Pei, Mohammad Minhajuddin, Angelo D'Alessandro, Rachel Culp-Hill, Kent A Riemondy, Austin E Gillen, et al. Venetoclax with azacitidine disrupts energy metabolism and targets leukemia stem cells in patients with acute myeloid leukemia. *Nature medicine*, 24(12):1859–1866, 2018.

[219] Eduardo H. P. Pooch, Pedro L. Ballester, and Rodrigo C. Barros. Can we trust deep learning models diagnosis? The impact of domain shift in chest radiograph classification. *arXiv preprint arXiv:1909.01940*, 2019.

[220] Kristina Preuer, Richard PI Lewis, Sepp Hochreiter, Andreas Bender, Krishna C Bulusu, and Günter Klambauer. Deepsynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics*, 34(9):1538–1546, 2018.

[221] Elizabeth Puddy and Catherine Hill. Interpretation of the chest radiograph. *Continuing Education in Anaesthesia, Critical Care and Pain*, 7(3):71–75, 2007.

[222] Nikaash Puri, Piyush Gupta, Pratiksha Agarwal, Sukriti Verma, and Balaji Krishnamurthy. Magix: Model agnostic globally interpretable explanations. *arXiv preprint arXiv:1706.07160*, 2017.

[223] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[224] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning with applications to medical imaging. *arXiv preprint arXiv:1902.07208*, 2019.

[225] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv:1711.05225*, 2017.

[226] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[227] Anil Rao, Joao M Monteiro, Janaina Mourao-Miranda, and Alzheimer's Disease Initiative. Predictive modelling using neuroimaging data in the presence of confounds. *NeuroImage*, 150:23–49, 2017.

[228] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

[229] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[230] 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.

[231] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.

[232] Laura Rieger, Chandan Singh, W. James Murdoch, and Bin Yu. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *Proceedings of the 37 th International Conference on Machine Learning*, 2020.

[233] Raquel Rodriguez-Perez and Jurgen Bajorath. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. *Journal of computer-aided molecular design*, 34(10):1013–1026, 2020.

[234] Raquel Rodriguez-Perez and Jurgenn Bajorath. Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values. *Journal of Medicinal Chemistry*, 63(16):8761–8777, 2019.

[235] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[236] Andrew Ross, Isaac Lage, and Finale Doshi-Velez. The neural lasso: Local linear sparsity for interpretable explanations. In *Workshop on Transparent and Interpretable Machine Learning in Safety Critical Environments, 31st Conference on Neural Information Processing Systems*, 2017.

[237] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[238] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[239] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Adversarially robust training through structured gradient regularization. *arXiv preprint arXiv:1805.08736*, 2018.

[240] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning (ICML)*, 2020.

[241] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv:1801.04381*, 2019.

[242] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[243] Rory Sayres, Ankur Taly, Ehsan Rahimy, Katy Blumer, David Coz, Naama Hammel, Jonathan Krause, Arunachalam Narayanaswamy, Zahra Rastegar, Derek Wu, et al. Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. *Ophthalmology*, 126(4):552–564, 2019.

[244] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[245] Monica Schenone, Vlado Dančík, Bridget K Wagner, and Paul A Clemons. Target identification and mechanism of action in chemical biology and drug discovery. *Nature chemical biology*, 9(4):232, 2013.

[246] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[247] Jacob Schreiber and Ritambhara Singh. Machine learning for profile prediction in genomics. *Current Opinion in Chemical Biology*, 65:35–41, 2021.

[248] Katharina Schultebraucks, Arieh Y Shalev, Vasiliki Michopoulos, Corita R Grudzen, Soo-Min Shin, Jennifer S Stevens, Jessica L Maples-Keller, Tanja Jovanovic, George A Bonanno, Barbara O Rothbaum, et al. A validated predictive algorithm of post-traumatic stress course following emergency department admission after a traumatic stressor. *Nature medicine*, 26(7):1084–1088, 2020.

[249] Andrew D Selbst and Julia Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, 12 2017.

[250] 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, 2017.

[251] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

[252] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[253] Ira Shavitt and Eran Segal. Regularization learning networks: deep learning for tabular datasets. In *Advances in Neural Information Processing Systems*, pages 1379–1389, 2018.

[254] Yuying Shi and Qianshun Chang. Efficient algorithm for isotropic and anisotropic total variation deblurring and denoising. *Journal of Applied Mathematics*, 2013, 2013.

[255] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[256] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[257] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[258] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[259] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[260] Chandan Singh, W James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*, 2018.

[261] Sumedha Singla, Brian Pollack, Junxiang Chen, and Kayhan Batmanghelich. Explanation by progressive exaggeration. In *International Conference on Learning Representations*, 2019.

[262] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[263] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[264] Le Song, Justin Bedo, Karsten M. Borgwardt, Arthur Gretton, and Alex Smola. Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, 23(13):i490–i498, 07 2007.

[265] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[266] Brett M Stevens, Courtney L Jones, Daniel A Pollyea, Rachel Culp-Hill, Angelo D'Alessandro, Amanda Winters, Anna Krug, Diana Abbott, Madeline Goosman, Shanshan Pei, et al. Fatty acid metabolism underlies venetoclax resistance in acute myeloid leukemia stem cells. *Nature cancer*, 1(12):1176–1187, 2020.

[267] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[268] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

[269] Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health AI. *Biostatistics*, 11 2019.

[270] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545 LP – 15550, oct 2005.

[271] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[272] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474*, 2019.

[273] Mukund Sundararajan and Ankur Taly. A note about: Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1806.04205*, 2018.

[274] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[275] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[276] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[277] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2019.

[278] Paul Takam Kamga, Federica Resci, Giada Dal Collo, Annalisa Adamo, Riccardo Bazzoni, Angela Mercuri, Massimiliano Bonifacio, and Mauro Krampera. Prognostic Impact of Notch Signaling in Acute Myeloid Leukemia (AML). *Blood*, 132(Supplement 1):5242–5242, 11 2018.

[279] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Learning global additive explanations for neural nets using model distillation. *arXiv preprint arXiv:1801.08640*, 2018.

[280] Minoru Tanaka, Yoko Hirabayashi, Takashi Sekiguchi, Tohru Inoue, Motoya Katsuki, and Atsushi Miyajima. Targeted disruption of oncostatin m receptor results in altered hematopoiesis. *Blood*, 102(9):3154–3162, 2003.

[281] Yi-Ching Tang and Assaf Gottlieb. Explainable drug sensitivity prediction through cancer pathway enrichment. *Scientific reports*, 11(1):1–10, 2021.

[282] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[283] Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*, 2017.

[284] Michael Tsang, Sirisha Rambhatla, and Yan Liu. How does this interaction affect me? interpretable attribution for feature interactions. *arXiv preprint arXiv:2006.10965*, 2020.

[285] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[286] Jeffrey W Tyner, Cristina E Tognon, Daniel Bottomly, Beth Wilmot, Stephen E Kurtz, Samantha L Savage, Nicola Long, Anna Reister Schultz, Elie Traer, Melissa Abel, et al. Functional genomic landscape of acute myeloid leukaemia. *Nature*, 562(7728):526–531, 2018.

[287] Saskia F van Vugt, Theo J M Verheij, Pim A de Jong, Chris C Butler, Kerenza Hood, Samuel Coenen, Herman Goossens, Paul Little, and Berna D L Broekhuizen. Diagnosing pneumonia in patients with acute cough: clinical judgment compared to chest radiography. *European Respiratory Journal*, 42(4):1076 LP – 1082, 10 2013.

[288] Maria de la Iglesia Vayá, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier Barber, Domingo Orozco-Beltrán, Francisco Garcia, et al. Bimcv covid-19+: A large annotated dataset of rx and ct images from covid-19 patients. *arXiv:2006.01174*, 2020.

[289] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald,

Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[290] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *arXiv preprint arXiv:1807.00199*, 2018.

[291] Linda Wang, Zhong Qui Lin, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10:19549, 2020.

[292] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[293] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[294] Ramsey M Wehbe, Jiayue Sheng, Shinjan Dutta, Siyuan Chai, Amil Dravid, Semih Barutcu, Yunan Wu, Donald R Cantrell, Nicholas Xiao, Bradley D Allen, et al. Deepcovid-xr: an artificial intelligence algorithm to detect covid-19 on chest radiographs trained and tested on a large us clinical dataset. *Radiology*, page 203511, 2020.

[295] Ethan Weinberger, Joseph Janizek, and Su-In Lee. Learning deep attribution priors based on prior knowledge. *arXiv preprint arXiv:1912.10065*, 2019.

[296] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

[297] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[298] Ho Yuen Frank Wong, Hiu Yin Sonia Lam, Ambrose Ho-Tung Fong, Siu Ting Leung, Thomas Wing-Yan Chin, Christine Shing Yen Lo, Macy Mei-Sze Lui, Jonan Chun Yin

Lee, Keith Wan-Hang Chiu, Tom Chung, et al. Frequency and distribution of chest radiographic findings in covid-19 positive patients. *Radiology*, page 201160, 2020.

[299] Pak Kin Wong, Fuqu Yu, Arash Shahangian, Genhong Cheng, Ren Sun, and Chih-Ming Ho. Closed-loop control of cellular functions using combinatory drugs guided by a stochastic search algorithm. *Proceedings of the National Academy of Sciences*, 105(13):5105–5110, 2008.

[300] Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[301] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv:1611.05431*, 2016.

[302] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, pages 10965–10976, 2019.

[303] Fuxun Yu, Zirui Xu, Yanzhi Wang, Chenchen Liu, and Xiang Chen. Towards robust training of neural networks by regularizing adversarial gradients. *arXiv preprint arXiv:1805.09370*, 2018.

[304] John R. Zech. What are radiological deep learning models actually learning? *medium.com*, 2018.

[305] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):e1002683, 11 2018.

[306] Hao Zhang, Bo Chen, Yulai Cong, Dandan Guo, Hongwei Liu, and Mingyuan Zhou. Deep autoencoding topic model with scalable hybrid bayesian inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[307] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

[308] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[309] Nan Zhang, Ying Chen, Shifeng Lou, Yan Shen, and Jianchuan Deng. A six-gene-based prognostic model predicts complete remission and overall survival in childhood acute myeloid leukemia. *OncoTargets and therapy*, 12:6591, 2019.

[310] Ran Zhang, Xin Tie, Zhihua Qi, Nicholas B. Bevins, Chengzhu Zhang, Dalton Griner, Thomas K. Song, Jeffrey D. Nadig, Mark L. Schiebler, John W. Garrett, Ke Li, Scott B. Reeder, and Guang-Hong Chen. Diagnosis of covid-19 pneumonia using chest radiography: Value of artificial intelligence. *Radiology*, In press.

[311] Xiaoyan Zhao, Yuan Li, and Haibing Wu. A novel scoring system for acute myeloid leukemia risk assessment based on the expression levels of six genes. *International journal of molecular medicine*, 42(3):1495–1507, 2018.

[312] Mingyuan Zhou, Yulai Cong, and Bo Chen. Augmentable gamma belief networks. *The Journal of Machine Learning Research*, 17(1):5656–5699, 2016.

[313] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[314] Dornoosh Zonoobi, Ashraf A Kassim, and Yedatore V Venkatesh. Gini index as sparsity measure for signal reconstruction from compressive samples. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):927–932, 2011.

[315] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.