New AI Frameworks for Real-World Clinical Prediction

Gabriel G. Erion

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

Doctor of Philosophy

University of Washington

2021

Reading Committee
Su-In Lee, Chair
Nathan J. White
Christopher Althoff

Program Authorized to Offer Degree:

Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

New AI Frameworks for Real-World Clinical Prediction

Gabriel G. Erion

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

In recent years, artificial intelligence (AI) has seen a string of successes for predicting clinical outcomes from eye disease to skin cancer to mortality. Clinical AI methods usually assume access to a large patient data matrix where rows are patients and columns are clinical variables, as well as a vector of labels representing each patient's outcomes. However, access to these data matrices is limited in many important clinical prediction tasks, making standard AI methods difficult to apply. We present methods to advance three broad areas of data-constrained clinical prediction. First, when AI models are deployed in fields like emergency medicine, providers may lack the time to gather all variables in the data matrix for input to an AI model when it is deployed. We addressed this problem with CoAI (Cost-Aware AI), which can automatically select highly predictive variables to make any AI model work within a desired clinical time constraint. Second, because most AI models require extremely large numbers of patient samples, it can be hard to train them when very little labeled data is available. We addressed this problem with sparse attribution priors, which enable a form of sparsity regularization in neural networks similar to sparse linear models and significantly improve performance with limited training data. Finally, when studying rare or emerging diseases, scientists may find that no outcome data is available at all to train predictive models. We addressed this problem with decoupled regression, an approach for synthesizing existing associations reported in the literature into full multivariate predictive models without ever using labeled training data. All three of these methods are general AI frameworks that are applicable to any predictive task within or outside of clinical medicine. We hope they will help to bring the impressive predictive power of AI methods into new clinical fields in which their application would have otherwise been impossible.

I would never have been able to complete this degree without the incredible support of my family and friends. My endless thanks to:

- My classmates, both in CSE and the UW MSTP, for their amazing support and for dragging me away from my computer to have fun when they knew I needed it.
- Jess, for having big dreams, for accomplishing them, and for inspiring me to do the same.
- Mom and Dad, for raising me to believe I could accomplish anything I wanted, and for raising me to be the kind of kid for whom "anything I wanted" meant a PhD!
- Most importantly to Lindsey, for being my rock throughout this PhD, and for agreeing to marry me.

Contents

Chapter 1

Introduction

Clinical risk scores, or mathematical models for predicting patient outcomes, have a long history in medicine. However, they are quickly becoming more prevalent than ever before. The risk score database MDCalc.com is one repository of such risk scores, providing a set of web-based calculators for predicting patient outcomes including death, recovery, hospital readmission, and disease progression [1]. MDCalc hosted 80 clinical risk calculators in 2013, and grew to over 500 calculators in 2019. Over the same period, the proportion of doctors in the United States who used MDCalc-hosted risk scores at least weekly climbed from 15 percent to over 65 percent [2]! In recent years, an increasingly large proportion of clinical risk scores have begun to use artificial intelligence (AI) methods, including those developed in the subfield of machine learning (ML), to make clinical predictions. AI models have made impressively accurate predictions when using images of skin and eyes to classify cancer [3] and diabetic retinopathy [4], waveform data such as electrocardiograms to classify heart arrhythmias [5], and comprehensive medical record data to predict patient diagnoses and surgical emergencies [6, 7]. Many of these models even exceed the performance of clinical experts' predictions on the same tasks. They promise to make clinical outcome prediction easier, faster, and more accurate for healthcare providers and to help many patients avoid death or serious injury.

Many of the impressive results enabled by machine learning and AI in clinical risk prediction stem from the combination of large, flexible predictive models with huge numbers of parameters and *big data*. Unlike simple predictive models such as linear models, whose performance tends to plateau as datasets grow larger, the complexity of these new models allows their performance to continually increase as the size of the dataset increases up to millions or billions of patient samples and thousands or tens of thousands of clinical measurements [8]. However, in many clinical situations where predictions must be made, it is difficult or impossible to access the types of huge datasets that make AI successful.

This thesis considers three specific areas where the big-data AI paradigm is incompatible with real-world clinical risk prediction scenarios and develops methods that improve the ability of AI methods to solve problems in these areas. Before my PhD, I was an EMT and later a medical student; I have entered information into Toughbooks in the back of an ambulance and discussed clinical risk scores with doctors as they assessed patient risk during my clinical rotations. Throughout my PhD, I encountered a wide variety of AI models for predicting patient outcomes and constantly asked myself "Could I see a doctor actually using this score after leaving the patient room? Would I myself take the effort to use it on a 911 call? Why or why not?" There are a huge number of hurdles to be cleared between the development of a clinical risk score and the day it improves a patient's diagnosis or treatment, and this thesis does not claim to solve them all. However, I believe there are real types of diseases and clinical encounters in which each of these tools could turn AI and machine learning from abstract novelties into the best tool for the job. Each chapter of this thesis considers one such tool and corresponds to one paper that has been published or submitted to a journal. I hope that the publication of these results expands, for even a few people, the idea of what machine learning can do in the field of medicine.

Chapter 2 addresses the problem of *costly features* and introduces a solution called *CoAI* (Cost-Aware Artificial Intelligence). AI-based risk scores are generally developed with the assumption that all of the *features* (i.e., clinical variables) in a clinical dataset will be known at prediction time, though acquiring all of these features in order to diagnose an individual patient may be impractical. Even sparse models, which select and use only a small number of features, do not account for the time or effort (more generally, *cost*) required to acquire those features. In clinical deployment, we may want to avoid gathering certain features due to their high cost. This is often the case in time- or resource-constrained fields such as emergency and critical care medicine due to time and attention limitations. We performed a survey of emergency medical services (EMS) providers in the Pacific Northwest and found very tight constraints on how many features could reasonably be gathered for an emergency medical risk score. We developed a method called CoAI (Cost-Aware Artificial Intelligence) to automatically build predictive models that fit within a clinically-relevant time or resource threshold for data gathering and evaluated it on trauma registry, intensive care unit, and outpatient clinic data. CoAI improved both the predictive performance and data-gathering cost of clinical risk prediction over existing clinical risk scores as well as other AI methods. The design of CoAI also allowed it to incorporate any predictive modeling strategy, improved training time, and improved the robustness of the model to realistic variations in data-gathering cost. CoAI has been released as open-source software; we believe it will help improve not just clinical risk prediction but predictive modeling in any field in which time or resource pressure necessitates highly efficient decision making.

Chapter 3 introduces the problem of *limited data* and introduces a solution framework called *attribution*

priors. In addition to relying on large numbers of features, AI-based risk scores also generally assume access to very large datasets to achieve high predictive performance. However, in many important clinical areas, data is unavoidably limited. For example, the largest patient dataset ever gathered for the rare genetic disorder alkaptonuria contained only 125 patients [9]. Cumulatively, up to 446 million people suffer from rare diseases like alkaptonuria worldwide [10]. Datasets for studying such diseases are often too small for complex AI models to exhibit good performance. When linear models perform poorly on small datasets, a common solution is to induce models to be sparse, such that most features have no effect on the model's output. This is often accomplished with methods like the Lasso penalty, but literature on how to achieve similar sparsity with AI models like neural networks is very limited. We introduce the framework of attribution priors, a set of methods for encouraging neural network models to assign importance to each input feature in a way that reflects our prior knowledge. Attribution priors can improve performance on small datasets by encouraging the network to assign low or zero importance to most features; this approach results in sparser models that are also more accurate than those created by other methods for sparse neural networks. Importantly, while my interest in attribution priors stemmed from their potential for limited-data modeling, they are a broad toolkit for incorporating prior knowledge into the learning process. My co-first-authors on the project, Joe Janizek and Pascal Sturmfels, contributed impressive results in drug response prediction and image classification that also form a large part of the paper. Subsequent work by other authors has used our attribution priors framework in clever ways to train networks that incorporate new kinds of biological and scientific knowledge that none of us predicted [11, 12]. We are very excited to see this progress continue!

Chapter 4 discusses the problem of *unavailable labels*, which we address with an algorithm called *decoupled regression*. Perhaps the most challenging modeling setting of all those we consider is one in which no labeled data is available at all. We may have access to a patient dataset, but not to information on the outcomes of interest in those patients. Such a situation can come about in many important clinical scenarios, including (1) when data sharing restrictions make it difficult for researchers to acquire labeled data, (2) when a new outcome is of interest in a population, but measurements of that outcome are not yet available to researchers, or (3) when many variables are hypothesized to be related to an outcome, but researchers do not have access to a dataset containing all variables and the outcomes measured simultaneously. All of these problems become particularly important when studying rare, new, or emerging diseases, for which no one researcher is likely to have all the data required to build a high-quality predictor of patient outcomes. Our approach to this problem, decoupled regression, is a method for meta-analysis of generalized linear models (GLMs). It is able to combine regression coefficients from several different studies into a single multivariate predictive model, whether the coefficients to be synthesized represent simple univariate associations or multivariate models themselves. While meta-analysis is a large area of research, very little such research has studied the problem of combining multiple GLMs to yield a single model. Decoupled regression has several benefits over the other methods that exist: it requires only regression coefficients and univariate summary statistics from each study, automatically supports any exponential family GLM with canonical link, does not require the user to have any outcome data of their own, and supports regularization and model selection. We found that decoupled regression performed well across a wide range of datasets and GLM types; in addition, we performed a meta-analysis to build an intensive care mortality prediction model and found that the ability to perform regularization and model selection greatly improved decoupled regression's performance in realistic scenarios where study coefficients exhibit noise and bias. We hope that decoupled regression will expand the range of scientific knowledge that can be meta-analyzed, synthesized, and ultimately, shared.

Each method presented in this thesis has been implemented as open-source software and shared publicly. My motivation in developing these tools always stemmed from an interest in their medical application, but every tool has applications far beyond medicine. Just as each tool was intended to broaden the ways in which AI and machine learning could be used in medicine, they have the potential to do the same in a wide array of fields. I hope that these methods help the next generation of predictive models become easier to train, deploy, aggregate, and share.

Chapter 2

CoAI: Cost-Aware Artificial Intelligence for Healthcare

The recent emergence of accurate artificial intelligence (AI) models for disease diagnosis raises the possibility that AI-based clinical decision support could substantially lower the workload of healthcare providers. However, for this to occur, the input data to an AI predictive model, i.e., the patient's features, must themselves be low-cost: efficient, inexpensive, or low-effort to acquire. When time or financial resources for gathering data are limited, as in emergency or critical care medicine, modern high-accuracy AI models that use thousands of patient features are likely impractical. To address this problem, we developed the CoAI (Cost-aware AI) framework to enable any kind of AI predictive model (e.g., deep neural networks, tree ensemble models, etc.) to make accurate predictions given a small number of low-cost features. We show that CoAI dramatically reduces the cost of predicting prehospital acute traumatic coagulopathy, intensive care mortality, and outpatient mortality relative to existing risk scores, while improving prediction accuracy. It also outperforms existing state-of-the-art cost-aware prediction approaches in terms of predictive performance, model cost, robustness to feature cost perturbations, and training time. These benefits stem from several unique strengths: First, CoAI uses axiomatic feature attribution methods that enable precise estimation of feature importance. Second, CoAI is model-agnostic, allowing users to choose the predictive model that performs the best for the prediction task and data at hand. Finally, CoAI decouples feature selection from model training, leading to faster and more flexible adaptation to new feature costs and prediction budgets. We believe CoAI will dramatically improve patient care in the domains of medicine in which predictions need to be made with limited time and resources.*

2.1 Main

Clinical risk prediction scores have a long history in medicine, and the number of such scores is rapidly increasing. The risk score database MDCalc.com hosted 80 clinical risk calculators in 2013 and over 500 in 2019, predicting adverse outcomes in conditions ranging from sore throat to heart failure [1]. In recent years, there has been an explosion of interest in using techniques from artificial intelligence (AI), including those developed in the subfield of machine learning (ML), to make clinical predictions. AI models use images of skin and eyes to classify cancer [3] and diabetic retinopathy [4], use waveform data such as electrocardiograms to classify heart arrhythmias [5], and use comprehensive medical record data to predict patient diagnoses and surgical emergencies [6, 7]. These models promise to make clinical outcome prediction easier and faster for healthcare providers; this possibility is especially important in areas such as emergency medicine and critical care, where providers' time and attention are at a premium.

However, both existing and AI-based clinical risk scores suffer from a common drawback; they generally assume that all of the *features* (i.e., clinical variables) in the training set are known at prediction time, though acquiring all of these features in order to diagnose an individual patient may be impractical. Even sparse models, which select and use only a small number of features, do not account for the time or effort required to acquire those features. In time- or resource-constrained fields such as emergency and critical care medicine, important features are often missing due to time and attention limitations. For example, from 1995 to 2009, Emergency Medical Service (EMS) providers in Washington State spent a median of just 16 minutes on the scene of trauma incidents [14]. The median time from EMS dispatch to arrival at the hospital was 48 minutes, just under the "golden hour" within which treatment affords the best chance of preventing death. These healthcare situations leave little time to gather data for feature-rich AI predictions. An alternative, *cost-aware* AI approach would account for real-world limitations by jointly optimizing for data gathering cost – e.g., time, effort, or money – as well as accuracy. Such a model could learn on massive datasets with many features and search for the optimal subset given any time, effort, or monetary budget. Most importantly, it would preserve the high accuracy of AI models while turning the heuristic process of feature selection into a principled optimization problem that the model can automatically solve.

We present a new AI framework, named CoAI (Cost-aware Artificial Intelligence; Figure 2.1), which calculates each feature's predictive power (Figure 2.1a) and uses expert annotations of feature cost (Figure 2.1b) to choose a highly-predictive set of features for any cost budget (Figure 2.1c). Given a new patient and a feature budget for prediction, CoAI can recommend which features to gather and make accurate predictions

*An earlier version of this work has been published as a preprint and is available as:
Gabriel G. Erion et al. "CoAI: Cost-Aware Artificial Intelligence for Health Care". *medRxiv*, published 2021, Cold Spring Harbor Laboratory. [13]. It has since been submitted for publication and is under review.

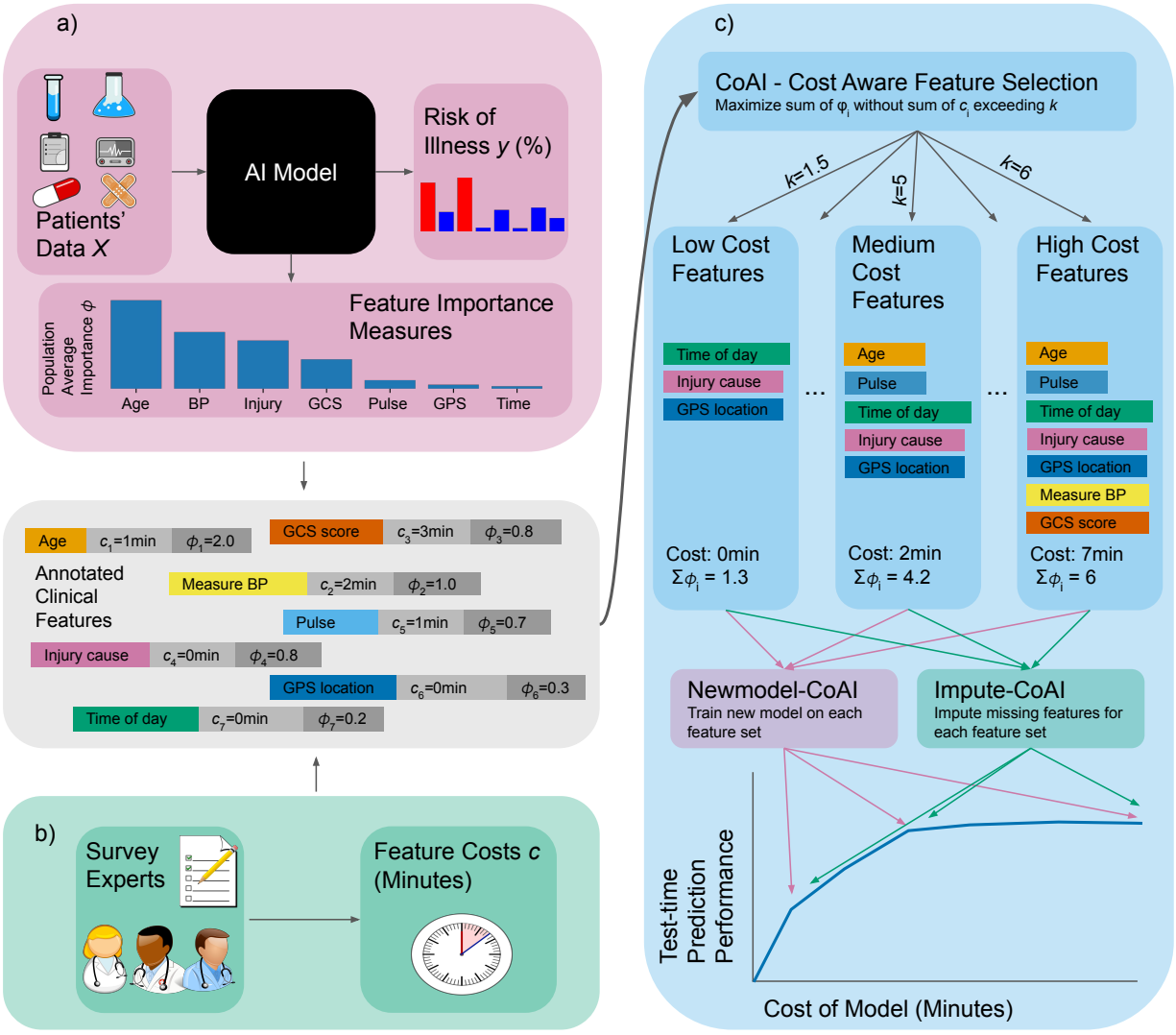
of patient risk given those features. Its main benefits include the abilities: (1) to quantitatively optimize the tradeoff between prediction performance and feature cost, yielding accurate *and* low-cost predictions; (2) to make *any* predictive model (e.g., deep neural networks, gradient boosted trees, etc.) cost-aware, dramatically increasing user choice; (3) to train the model significantly more efficiently and with less hyperparameter tuning than existing cost-aware prediction approaches; and (4) to maintain higher robustness than other methods to fluctuations in feature cost or the data-gathering budget. We have released CoAI as open-source software compatible with the popular Scikit-Learn API for training AI models [15] [†]. We compare CoAI against existing clinical risk scores and recent AI methods for cost-aware prediction, including cost efficient gradient boosting (CEGB) and a reinforcement-learning (RL) based method called "classification with costly features" (CWCF) [16, 17, 18] on three clinical tasks (Figure 2.2; Methods 2.4.1).

Our first cost-aware prediction task involves acute traumatic coagulopathy (ATC) in our trauma dataset – 14,000 emergency room visits and 46 features from the trauma registry of Harborview Medical Center, an urban Level-I trauma center (Figure 2.2a). ATC is an increased bleeding tendency involving anticoagulation and clot breakdown affecting up to 30% of severely-injured trauma patients [19]. It is associated with acute kidney and lung injury, increased transfusion needs, multiple organ failure and an 8-fold increased risk of early death [20, 21]. Trauma patients with ATC require complex care and rapid mobilization of hospital resources including massive blood transfusion protocols and surgical teams [22]. ATC diagnosis is currently based on coagulation testing in the hospital, which delays this time-critical diagnosis and complex healthcare response [23]. Therefore, our goal is to identify trauma patients at high risk of ATC as early as possible before arrival at the hospital to enable faster hospital-based life-saving interventions. Triage is only one of many tasks EMS providers must perform in trauma responses, so we aim to minimize the time required to gather input features. We selected prehospital features from the Harborview trauma registry, identified the data-gathering cost in minutes for each feature by surveying local experienced EMS providers (Figure 2.3f; Methods 2.4.2; Supplementary Figure 2.6; and Supplement Sections 2.5.2-2.5.3), trained a CoAI ATC prediction model, and compared its cost and predictive performance with existing tools that predict ATC from prehospital data [24, 25]. One such model, the Prediction of Acute Coagulopathy in Trauma (PACT) score, required up to an estimated ninefold more data-gathering time than EMS providers reported being willing to spend during trauma responses, indicating the need for cost-aware modeling [24].

We also examine the problem of in-hospital mortality prediction in critical care patients in our "ICU dataset" – 140,000 patient stays and 43 features from intensive care units (ICUs) across the United States (Figure 2.2b). ICU mortality is an important prediction target because it is (1) prevalent, with mortality rates in United States ICUs as high as 19 percent (2) costly, with ICU expenditures constituting 14 percent of hospital costs, and (3) highly variable across patients and hospitals even after adjusting for baseline patient characteristics [26]. For this task, we define cost simply as the *number of features* in the model. This is motivated by the fact that the mortality risk scores with the highest accuracy, such as the Acute Physiology and Chronic Health Evaluation (APACHE) model, are considered impractical for clinical use because they require a large number of features [27]. Models designed to be more efficient for clinical use, such as the Sequential Organ Failure Assessment (SOFA) and quick SOFA (qSOFA) scores, use as few as 3 features but suffer reduced performance as a result [28]. Our goal is to provide a single method that optimally trades off cost with performance and can provide accurate predictions using any number of features deemed clinically feasible.

Finally, we examine 10-year mortality prediction in an "outpatient dataset" from the long-running National Health and Nutrition Examination Survey (NHANES) – 13,000 outpatients and 35 features in outpatients across the United States (Figure 2.2c) [29]. Here, the goal is to minimize the *financial* cost in dollars of the data used by the model as estimated using Medicare fee-for-service data (Methods 2.4.3; Supplementary Figure 2.7) while preserving prediction accuracy. This task uses a unique patient sample representative of the United States population. We choose such a dataset because (1) mortality prediction is a ubiquitous clinical task; a national survey found that internal medicine physicians were asked to predict patient lifespan roughly once per month but felt ill-prepared to do so [30], and (2) a model that is applicable to all 480 million annual primary care visits in the United States [31] is an important case in which to lower the financial cost of risk scores. While long-term mortality scores have been developed for many specific diseases and patient subpopulations [32, 33, 34], we are not aware of a commonly-used outpatient mortality risk score applicable

[†] `https://github.com/suinleelab/coai`



d)

	Training			Test	
Newmodel-CoAI	**Train** base model Calculate **importance** φ_i	**Select** feature sets at many possible budgets	**Train** models at many possible budgets		Given a budget, **predict** with best available pretrained model
Impute-CoAI		**Train** imputation model		Given a budget, **select** feature set	Impute missing features and **predict** with base model

Figure 2.1: Overview of CoAI framework. Clinical features are annotated based on two different sources. (a) The *importance* of each feature is calculated by training an AI predictive model on the full data and applying additive feature attribution methods to that model. (b) By surveying clinical domain experts, the *cost* of gathering each feature is estimated. We consider time cost (minutes) and financial cost (dollars), though CoAI supports any numeric quantity. (c) (top) The CoAI algorithm takes as input all features, costs, and importance values and selects appropriate feature subsets for any cost budget. Newmodel-CoAI (NM-CoAI) trains a new AI model for each feature subset and cost budget, while Impute-CoAI (I-CoAI) measures the selected features at a given budget and imputes missing features. (bottom) Training CoAI models with multiple budgets results in a cost-performance tradeoff curve, as larger budgets allow for gathering more features, which leads to higher-accuracy predictions. (d) NM-CoAI performs feature selection at train time and pretrains models for each budget. I-CoAI defers feature selection to test time.

to the general primary-care population. We hypothesize this is due in part to the prohibitive expense of gathering data for routine mortality prediction, and hope to show that accurate, low-cost predictions can be made in this setting.

Across all three tasks, CoAI consistently improved predictive performance and lowered cost relative to both existing clinical risk scores and existing AI-based methods. CoAI can also improve training time and robustness to shifts in feature cost. It bridges the gap between AI-based predictive models and the real-world constraints of clinical practice by ensuring that predictive models do not impose undue burden on their users. We believe this work will improve the accuracy of clinical risk predictions while ensuring that such predictions are made efficiently enough to have a real-world impact on patient care.

2.2 Results

2.2.1 CoAI Framework

We present the CoAI (Cost-aware AI) framework for cost-aware prediction. Because of the wide array of prediction tasks and modeling strategies used in developing clinical risk scores, CoAI can be applied to any predictive model (called the *base model*) to make it cost-aware (Figure 2.1; Methods 2.4.4). CoAI takes as input a training data set, consisting of patient data X with n samples and m features, prediction labels y across patients, *costs* c_i for measuring each feature i, and a *budget* k representing total acceptable cost for a predictive model. The goal of CoAI is to select a specific feature set S, with total cost no greater than k, that yields the best predictive performance given the budget.

This task is computationally challenging because there are a huge number of such subsets for any practical number of features and the exact predictive value of a feature set is unknown without training the model on that set. Previous approaches based on reinforcement learning (RL) attempt to directly search the exponentially large space of all possible feature sets, while others, such as decision tree-based approaches, approach the same problem with a greedy search heuristic [16, 17]. These methods yield approximate solutions to an intractable problem; CoAI takes a different approach and finds an exact solution to a simpler problem. The key simplifying assumption made by CoAI is that a single quantitative measure of predictive power ϕ_i can be defined for each feature i,, such that the predictive power of a feature set S is equal to $\sum_{i \in S} \phi_i$. After making this assumption, selecting the feature set S that maximizes $\sum_{i \in S} \phi_i$ subject to $\sum_{i \in S} c_i \leq k$ is a knapsack problem that can be efficiently and exactly solved [35, 36].

The quality of our solution depends on whether we can find additive feature importances ϕ_i that are a good measure of predictive power. Recent work argues that Shapley values of a model's loss function are in several respects the optimal additive measure of feature importance [37].

$$\phi_i = \frac{1}{m} \sum_{T \subseteq M \setminus \{i\}} \binom{m-1}{|T|}^{-1} \left(\mathbb{E}\left[\ell(y, f_T(X_T))\right] - \mathbb{E}\left[\ell(y, f_T(X_{T \cup \{i\}}))\right] \right) \tag{2.1}$$

where f is the model, M is the set of all features, ℓ is the loss function, and $f_T(X_T)$ represents the model's output when only the values of feature set T in the dataset are known (more detail in Methods 2.4.5 and [38]). When calculated on the global loss across an entire dataset, these Shapley Additive Global Explanation (SAGE) values are the only set of additive values that satisfy a set of desirable properties, including symmetry with respect to perfectly correlated variables, zero attribution for uninformative variables, and monotonicity – the property that, for features a and b such that hiding feature a will always increase the model's loss more than feature b across all possible sets of other known features, $\phi_a \geq \phi_b$ ([38], Properties 1,2, and 4). In addition, while any such additive importance measure is an approximation – for example, feature y may be much less useful when a correlated feature x is already known than when it is not – SAGE values are the best additive approximation of feature value for a given model with respect to mean squared error across all possible combinations of missing features ([38], Equation 8). Empirically, Shapley value attributions of the loss have been shown to be effective for feature selection, though such research has not to our knowledge incorporated feature cost [38, 39].

Finally, Shapley value attributions, including SAGE, can be calculated for any model, making CoAI a model-agnostic method and increasing user flexibility. While Shapley value attributions can be computationally expensive to calculate, these explanation methods usually estimate importance for each sample in the dataset.

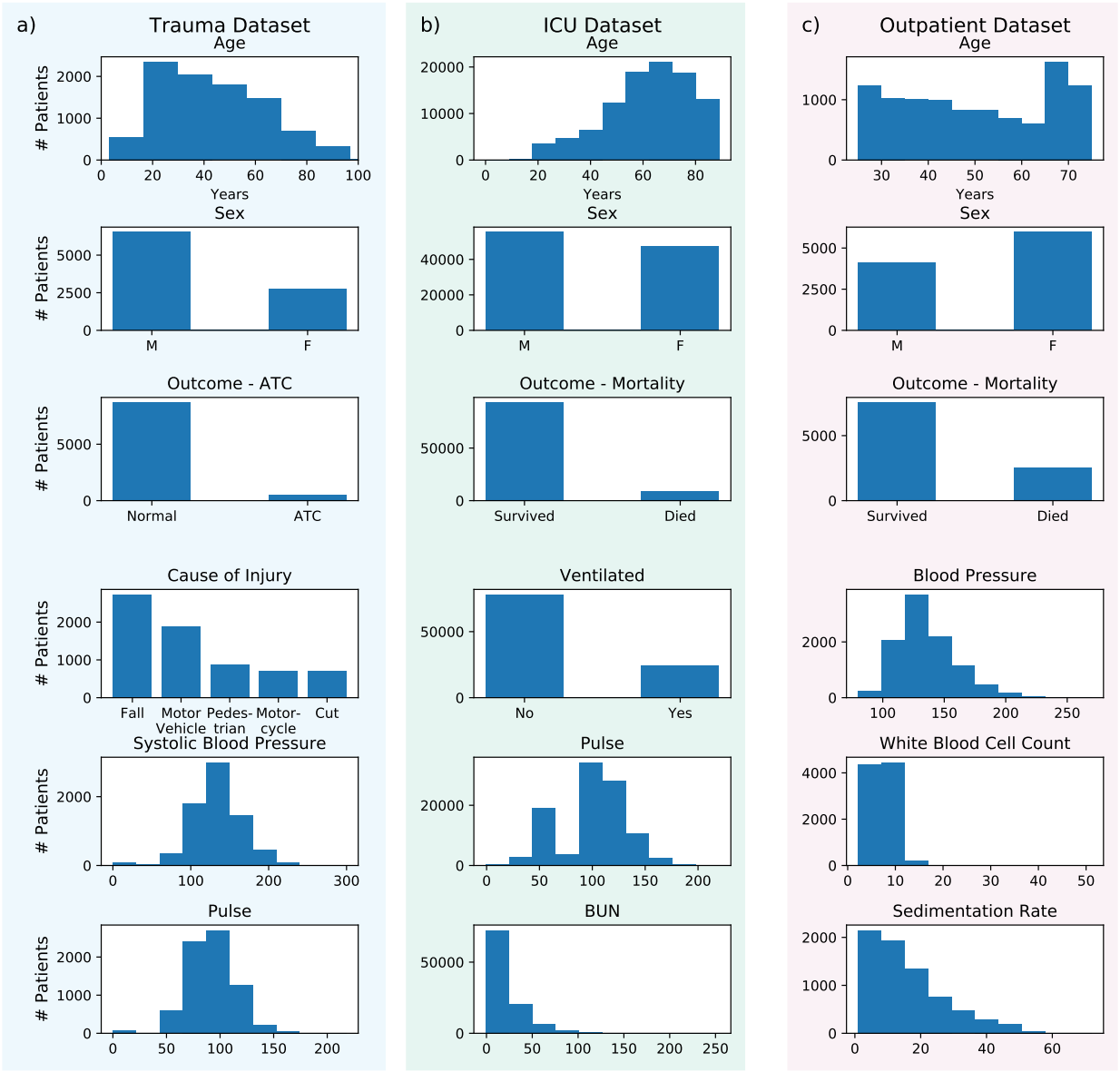


Figure 2.2: Histograms providing statistics for the (a) trauma, (b) ICU, and (c) outpatient datasets. The first three rows for each dataset show the distribution of age, sex, and outcome of interest for each dataset, respectively. The bottom three rows show the distribution of the next three most important features in each dataset, as measured by feature importance on a random train/test split (Methods 2.4.5). Notably, the trauma dataset and ICU dataset have clear age bias (younger patients are more likely to have traumatic injuries and older ones are more likely to end up in the ICU), while the outpatient dataset has a more uniform distribution as it was designed to be representative of American adults.

Because we only need global importance across the entire dataset, global methods like SAGE provide accurate estimates of importance in much less time. All calculations in this paper converged quickly. However, if explanations on very large datasets are required, model-specific Shapley value methods and even faster approximations exist for deep, tree, and linear models, among others [40, 39, 41, 37].

We calculate ϕ_i by training an instance of the base model on all features in X and setting ϕ_i equal to the model's SAGE values. We then can use existing specialized knapsack solvers to select the feature set S [36]. There are now two possible approaches; in the Newmodel-CoAI (NM-CoAI) approach, we select S at training time, then train a new model on the features in S. In the alternate approach, Impute-CoAI (I-CoAI), we calculate ϕ_i and train a feature imputation model on X, but we do not select S. At test time, we look at the cost vector, select S, impute the features not in S, and feed them to the full-feature model (Figure 2.1). In general, the former approach should be slightly more accurate [42], but because Impute-CoAI does not use costs until test time, it allows for quick adaptation to new cost vectors or data-gathering budgets (Section 2.2.3). Newmodel-CoAI and Impute-CoAI can also be combined for best performance – new models are trained given the feature sets that optimize the training-time costs, but Impute-CoAI is used if costs change at test-time. We separate the two methods for clarity of exposition, but recommend combining them this way in practice; all results shown for CoAI use Newmodel-CoAI, except in Section 2.2.3 where we study changing costs and both methods are displayed. We also demonstrated that knapsack solvers performed better than alternative approaches, such as greedy selection and recursive feature elimination, to find the feature set S given ϕ_i and c_i (Methods 2.4.6; Supplementary Figure 2.8).

2.2.2 CoAI improves cost and performance of clinical predictions.

We evaluated CoAI against existing clinical models and other cost-aware AI methods by plotting its predictive performance (area under the ROC curve, AU-ROC; higher is better) across a range of measurement budgets, resulting in a cost-performance tradeoff curve (Figure 2.3a-c). Thus, the "CoAI (GBM)" curve corresponds to 100 trained GBMs, one for each of many data-gathering budgets. We used gradient boosting machines (GBMs), logistic regression, and neural networks (multi-layer perceptrons, or MLPs) as the base model (Methods 2.4.7). Existing clinical risk scores are shown on the same plots as points with a fixed model cost and performance. Other AI methods are shown as lines, also representing multiple trained models in a single curve. CoAI significantly outperforms existing clinical and AI methods in most cases across all datasets, and is never outperformed in any dataset.

Comparison with clinical risk scores

To determine how CoAI could improve prediction of ATC in the trauma dataset, we compared it to the Prediction of Acute Coagulopathy of Trauma (PACT) score (Figure 2.3a; Methods 2.4.8; Methods 2.4.9) [24]. PACT is a multivariable logistic regression developed for prehospital ATC prediction. It uses the following prehospital features: patient age, presence of prehospital CPR, presence of prehospital intubation, injury mechanism, Glasgow Coma Score, and shock index (first prehospital pulse/systolic blood pressure). In our survey of EMS providers, total time cost incurred to obtain all PACT features was 7.4 minutes (Figure 2.3f).

We compared ROC plots of PACT to those of CoAI at several clinically important points along the cost-performance tradeoff curve (Figure 2.3d). For the same time cost as the PACT score (7.4 minutes), CoAI performs almost identically to a cost-unconstrained model (0.82 AU-ROC) and exceeds PACT score performance (0.80 AU-ROC). We also determined a realistic time budget using our survey of EMS providers (Figure 2.3f), who reported being willing to spend 50 seconds using a predictive risk tool on average. This tightly constrained budget is about ninefold less time than the PACT score requires, but the performance of CoAI within this budget (0.80 AU-ROC) still matches PACT's performance. Importantly, CoAI's prehospital prediction performance compares favorably to existing *post*-hospital admission ATC models; a previous study of AI models for ATC achieved AU-ROCs from 0.83 to 0.86 using vital signs, blood gas measurements, and lab values gathered *after* patients entered the hospital [43]. CoAI attains similar performance using only tightly time-constrained prehospital data. We also evaluated performance at three specific binary prediction thresholds by fixing sensitivity to levels used in prior papers on coagulopathy prediction – 73% [24], 60% [25], and 75% [44] – and assessing the resulting specificity. PACT achieved specificities of 73.5%, 81.5%, and 72.5% at these thresholds, while CoAI improved specificity to 77.5%, 87.5%, and 77% respectively.

For the ICU mortality prediction dataset (Figure 2.3b), the APACHE IVa score is known for its high accuracy; however, is difficult for clinicians to use at the bedside because it requires 27 features to be gathered (Methods 2.4.8; Methods 2.4.9). Conversely, the qSOFA score uses only 3 features but is much less accurate. CoAI outperforms qSOFA using only a single feature (admission diagnosis, AU-ROC 0.75) and outperforms APACHE IVa using less than half as many features (AU-ROC 0.88). CoAI also outperforms the related APACHE III and APS scores at much lower model cost. The results exhibit a clear trend in model complexity as well. The APACHE IVa score, which incorporates the largest number of interactions and nonlinearities, comes closest to CoAI's performance, while the qSOFA score, which is very simple and adds together three binary yes/no variables, does the worst relative to CoAI. In a sense, qSOFA aims for a different goal than CoAI by achieving a lower point on the cost-performance curve in exchange for ease of computation. We also evaluated performance at the sensitivity corresponding to the binary threshold most often used for the qSOFA score (2 points). CoAI with 3 features improves specificity from 0.8 to 0.9 at the qSOFA sensitivity, outperforming qSOFA but not the APS and APACHE scores. CoAI with 27 features outperforms all scores, raising specificity to 0.97 from APACHE IVa's 0.95.

Comparison with AI methods

In addition to clinical risk scores, we compare CoAI to AI methods representing two important categories of existing cost-aware AI methods: (1) cost-efficient gradient boosting (CEGB), a popular and effective method for making cost-aware predictions with decision trees [16], which applies a fixed per-feature penalty whenever the tree model splits on a given feature for the first time, and (2) a reinforcement learning method called classification with costly features (CWCF) [17, 18], which penalizes an agent for selecting costly features and rewards it for producing correct classifications (Methods 2.4.10-Methods 2.4.12). Overall, CoAI consistently outperforms other AI methods (Figure 2.3a-c). The best CoAI model in each plot has a significantly higher mean AUC, averaged across all possible budgets, than all other models, with $p < 10^{-2}$ and $t > 3.52$ by paired-samples t-test for all comparisons (Methods 2.4.13).

For all datasets, cost-effective models reduce costs significantly without sacrificing performance, with CoAI and CEGB both plateauing in performance about halfway through the range of possible budgets or earlier. CWCF's reinforcement learning (RL) approach also consistently underperforms other methods. Although RL has exhibited strong performance in binary classification [17, 18], it suffers in risk-stratification settings that use ROC and similar metrics because it produces hard classifications rather than a continuous *ranking* of patients by risk. We adapted CWCF to produce continuous outputs and ran fewer replicates to accommodate its slower runtime, but performance was still very low (Methods 2.4.12).

The results in Figure 2.3a-c show several specific benefits of CoAI's design relative to other cost-aware learning methods. First, CoAI is *model-agnostic*, i.e, it can be used with any predictive model. This allows us to display results for several different base models; while GBMs perform best in the cases we examine, in areas where linear or deep models are preferred, CoAI is applicable with little to no modification. For all three model classes, performance plateaus at a similar rate, indicating a generalizable ability to select and use high-yield features.

Second, CoAI can easily use features that come in *groups* that each have a single acquisition cost. This situation is extremely common in AI tasks where features are redundantly encoded – such as nonlinear transformations of features or one-hot encoding – as well as in clinical medicine where many features (e.g., urine tests for blood and pH) can be acquired with a single lab test (urinalysis). CoAI naturally handles these situations because feature importance measures ϕ's are additive and can be easily summed to yield *group* importances (Methods 2.4.14). In the outpatient data, 27 exam findings and lab tests (groups) gave rise to 35 features, which expanded to 118 features after one-hot encoding. While each model could access all features, only CoAI could correctly place costs at a group level during training. CEGB and CWCF could not incorporate knowledge about grouped costs during training, so when they were evaluated on grouped costs at test time their performance was worse than CoAI. CWCF did not report the features gathered, making it difficult to estimate the effect of grouped costs. However, we calculated an upper bound on its performance at each cost and still found it underperformed CoAI (Figure 2.3a-c, Methods 2.4.12 and Methods 2.4.13).

Third, CoAI benefits from a non-greedy feature selection process. While CEGB exhibits high performance overall, when it chooses to use a new feature, it is balancing the cost of gathering the feature against the utility of a *single* split on that feature – regardless of whether further splits might improve performance. For

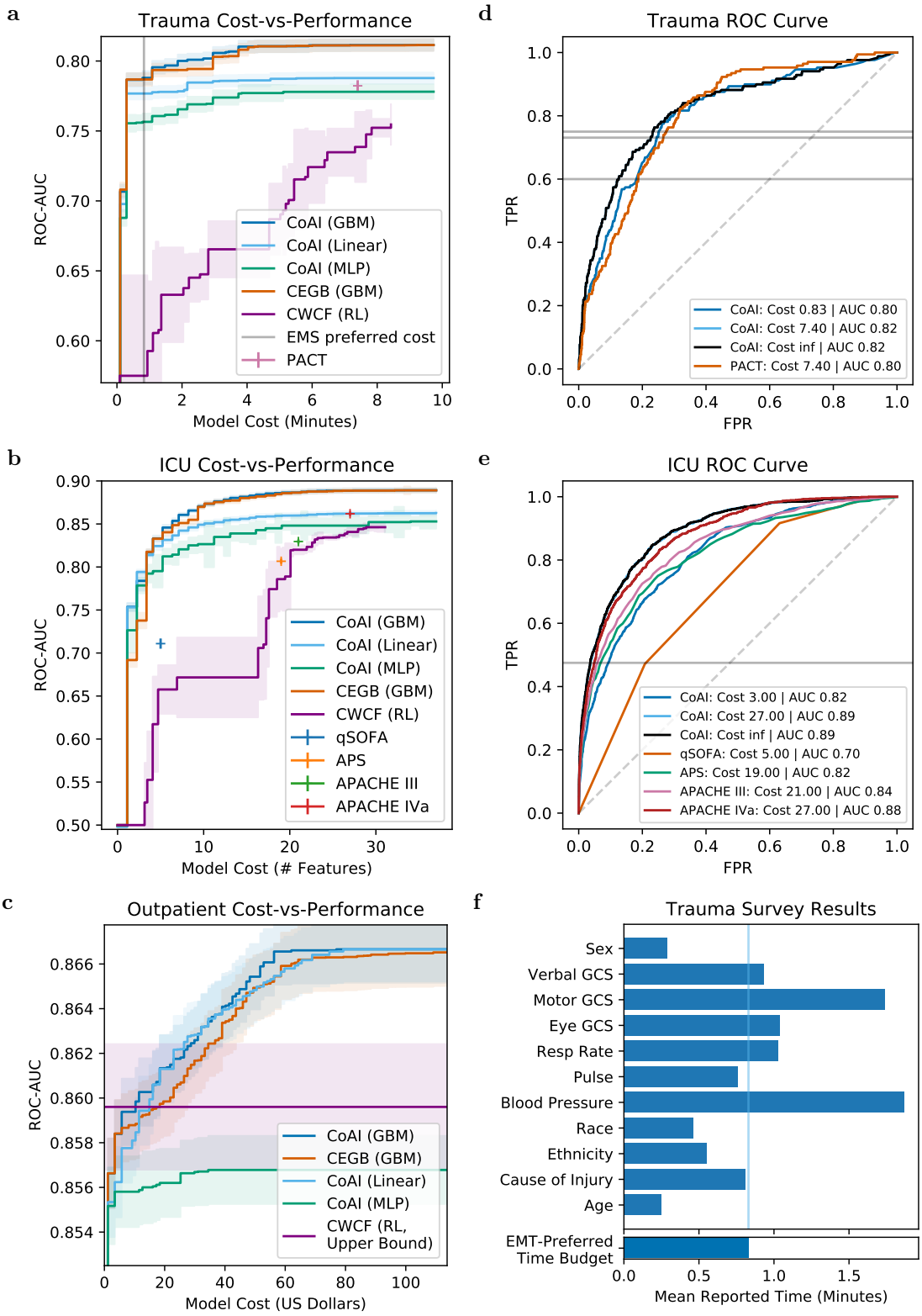


Figure 2.3: CoAI improves prediction performance and model cost over existing clinical models and AI methods. a-c) Cost-versus-performance plots for all methods on all datasets. Lines are the mean and shading is a 95 percent confidence interval over multiple train-test splits. d-e) Receiver Operating Characteristic curves for trauma and ICU datasets. Horizontal lines represent sensitivity points of clinical interest based on prior literature. In d) and e), the higher-cost CoAI model (7.4 minutes and 27 features) overlaps indistinguishably with the cost-unconstrained model. f) Feature costs for the trauma dataset. Zero-cost features are not shown; see Supplementary Figure 2.6 for more detail and costs for other datasets. We also report the EMS-preferred data-gathering cost for the trauma data.

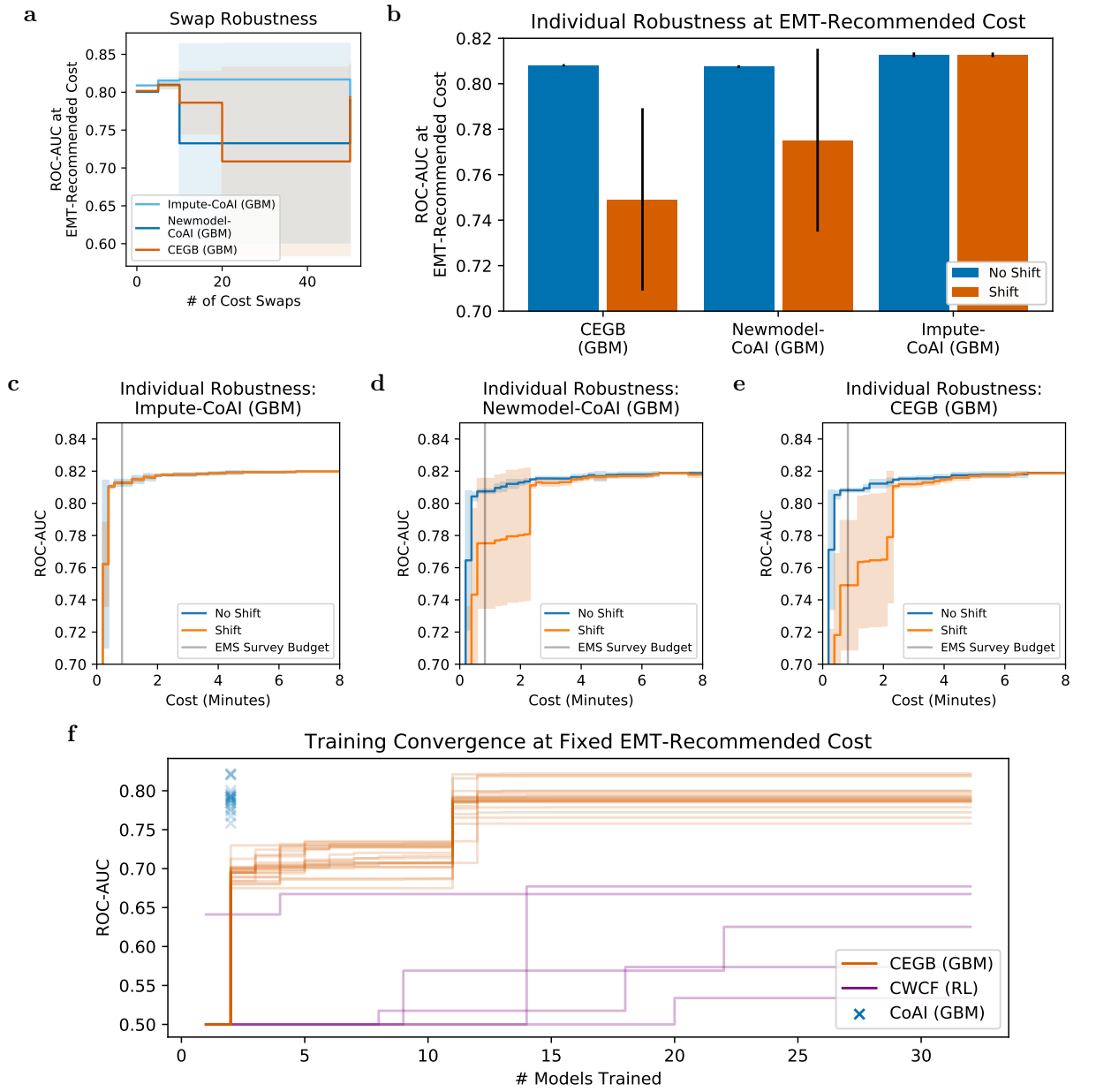


Figure 2.4: CoAI provides improved robustness and training complexity over competitor methods. a) Test performance of cost-aware AI methods as increasing numbers of swaps are applied to the cost vector between train and test time. Lines are means and shades are 95 percent CIs over random swap seeds. b-e) Cost versus test performance of cost-aware AI methods for all combinations of 5 complete cost vectors from unique individuals in our EMS survey. Blue lines and shading indicates a cost-performance curve from training and testing on the same cost vector and orange indicates using different vectors at train and test time. Results are averaged over all possible pairs of train/test vectors, with shading representing 95 percent CIs. Bar plot in b) summarizes performance at the EMT-recommended cost for each model. f) Training time, in number of models trained, for cost-aware AI methods. Methods that require no tuning are shown as single points, whereas methods that require parameter tuning to balance cost and performance are lines, with performance of the best model found increasing as more parameters are searched. Results from all 50 runs are overlaid (only 5 runs were performed for CWCF).

a simple tree structure and binary feature, one split might represent the full value of the feature. However, for a high-dimensional categorical feature like ICU admission diagnosis, one split could never capture the full value of gathering the feature and potentially using it in future splits. This is exactly what happens in the left hand of Figure 1b; CoAI is able to "look ahead" by seeing how important each feature was to a fully-trained model and select admission diagnosis as the first feature, resulting in 0.75 AU-ROC, while CEGB does not select admission diagnosis and achieves a lower AU-ROC (<0.7).

2.2.3 Decoupling feature importance, cost, and prediction allows more flexible modeling

CoAI enables robustness to shifting feature costs

Cost-performance curves like those in Figure 2.3a-c assume that feature costs are constant between training and test time; however, in practice costs are likely to differ from one hospital, region, or time period to another. All cost-aware AI methods we are aware of incorporate costs directly into model training: the structure of cost-aware decision trees depends on costs because they choose splits based on cost, and RL methods use a joint model for feature selection and prediction. However, CoAI decouples the feature selection step from the model training step, creating an opportunity for improved robustness. Our I-CoAI model (Section 2.2.1, Figure 2.1) can make test predictions using any cost vector without retraining because no cost vector is required at training time, and any cost vector can be used at test time.

We tested I-CoAI on the trauma data with two types of experiments. In both cases, we used iterative linear imputation from Scikit-Learn (a variant of the popular MICE method) as the imputation strategy [15, 45]. However, we note that CoAI is compatible with any imputation method. Our first experiment examined swap robustness, in which we trained with the same cost vector from the experiments in Section 2.2.2 but permuted elements of the cost vector at test time (Figure 2.4a). We then tested performance at the EMT-recommended budget of 50 seconds. I-CoAI adapted its predictions to the shifted test costs; other methods like CEGB and NM-CoAI did not support shifting costs. We adapted them by training a range of models across the cost-performance curve with the costs available at training-time, re-calculating each model's cost with the new test-time costs, then reporting the best predictive performance that satisfied the cost constraint. This was the best possible adaptation short of retraining, which may often be impossible upon deployment. As increasingly large numbers of swaps are applied, the performance of CEGB and NM-CoAI without retraining quickly drops. The performance of I-CoAI, however, is unaffected.

In our second set of experiments, we examined robustness to more realistic cost perturbations, which we call individual robustness. Here we disaggregate the costs reported in our EMS trauma dataset survey into individual responses and randomly select 5 individual results that reported complete cost vectors. We then train with one cost vector and test on another one (with the same CEGB and NM-CoAI adaptations as before) for all possible pairs of train-test assignments. We display the average cost-performance curves for the case when the train and test cost vectors are the same and when they are different in Figures 2.4b-d. CoAI and CEGB's performance drops dramatically, especially at low target costs, when the train and test cost vectors differ; in contrast, I-CoAI demonstrates almost identical performance when the train and test cost vectors are the same and when they are different. In addition, I-CoAI performs at least as well as NM-CoAI and CEGB even when no shift is present. This is surprising in light of previous literature that argues that retraining on feature subsets usually outperforms imputation [42] and would be interesting to investigate further. Overall, the choice of whether to use I-CoAI or NM-CoAI is made easier by the fact that both methods can coexist. Training an imputation model as well as new base models at a range of possible cost budgets allows the user to decide whether to use I-CoAI or NM-CoAI at test time, based on validation performance of each model and whether or not cost shift is present.

CoAI reduces complexity of cost-aware model training

Here, we demonstrate that CoAI can also train faster at a given budget than other AI methods. In particular, we want to minimize what we call *training complexity* – the number of base models that must be trained – in order to achieve a *class-optimal* model – the best model that can possibly be trained by the cost-aware algorithm. All cost-aware models require a tuning parameter to control the tradeoff between accuracy and cost. For CoAI, this parameter is the budget itself. Given a fixed target budget, CoAI always yields a class-optimal

model in two training rounds (the base model and the at-cost model). For CEGB and CWCF, however, the tuning parameter is unitless (representing the ratio between model cost and loss in the optimization objective). Given a fixed target budget, this requires blind tuning until a good enough model is found that fits within the budget.

We tested the training complexity of all models on all train-test splits of the trauma dataset, where we attempted to maximize performance at a single budget (the EMS provider-preferred time cost of 50 seconds) with each model (Figure 2.4d; Methods 2.4.15). Blind tuning on CEGB with binary search requires training over 5 times more models than CoAI to reach a similar level of performance (11-12 total models). CWCF takes a large number of trainings to yield even a small increase in performance and never reaches the same level of performance as CoAI or CEGB ($\geq$ 32 total models). Only CoAI is able to offer high predictive performance with a low number of model trainings.

2.2.4 CoAI reveals high-yield features and dynamics of feature importance over time

We analyzed the order in which CoAI selected features to understand how it differs from other clinical risk scores. Figure 2.5 shows how feature rankings differed between CoAI and existing clinical models for the ICU and trauma datasets. For CoAI, the top features are listed in the order in which they were first added to the model as the budget was gradually increased. For each clinical risk score, the top features are listed in order of importance; for PACT, importance is measured by standardized regression coefficient. The qSOFA score weights all variables equally, so the ordering is arbitary. APACHE IVa is a proprietary model for which it is difficult to obtain per-feature importance values, so we trained a GBM model to predict APACHE IVa scores (Spearman $R = 0.98$) and calculated its feature importance importances to generate a feature ranking (Methods 2.4.9) [38].

For the ICU data, CoAI and existing clinical models rely on different subsets of features (Figures 2.5a and 2.5b). Although CoAI and APACHE both rely on admission diagnosis and age, CoAI ranks ventilation, heart rate, FiO2, temperature, and WBC count, higher than APACHE, while ranking admission source, BUN, respiratory rate, and hematocrit lower. The qSOFA model uses a small number of features, most of which are also used by CoAI, although CoAI also relies on many features not chosen by qSOFA. Notably, the higher-ranked CoAI features tend to be baseline information – age, diagnosis, and ventilation status – rather than specific vital signs. A similar situation arises with the PACT score in the trauma dataset; many PACT features are also important in the CoAI model, but CoAI prioritizes inexpensive data such as knowledge of dispatch information and which procedures were performed before relying on the many vital signs used in PACT (Figure 2.5c). In particular, the intubation and CPR procedures are ranked more highly by CoAI than by PACT.

A unique aspect of CoAI is the opportunity to examine how changing budgets might affect the value of individual features. To investigate this, we examined the effects of changing time budgets on feature importance for the trauma CoAI models. The heatmap in Figure 2.5d displays the importance of each feature as a function of model cost as the budget k increases. This analysis reveals the dynamics of important features over time. For example, when very little time is available, dispatch and procedure information dominate the prediction (top row). Dispatch features include the patient's geographical location and level of response e.g., Advanced Life Support or Basic Life Support, and procedures include CPR and intubation. This result agrees with recent data suggesting that prehospital procedures have high predictive value for predicting the need for hospital interventions such as massive transfusion [46]. However, when time budgets increase other features, such as vital signs, gain more significance and the importance of the earlier demographic features tends to decline. Occasionally, features like pulse are removed from the model entirely. Further, a model that adds features in a fixed order (e.g., greedy and recursive feature elimination methods, see Methods 2.4.6 and Supplementary Figure 2.8) cannot make these sophisticated time and information tradeoff choices and would lose performance as a result. These results show that CoAI's ability to train models within any budget is valuable not only for flexibility in deployment, but also as a way to better understand the predictive value of each patient feature. Further analysis of feature importance, as well as importance heatmaps for other datasets, are shown in Supplementary Figures 2.9-2.13.

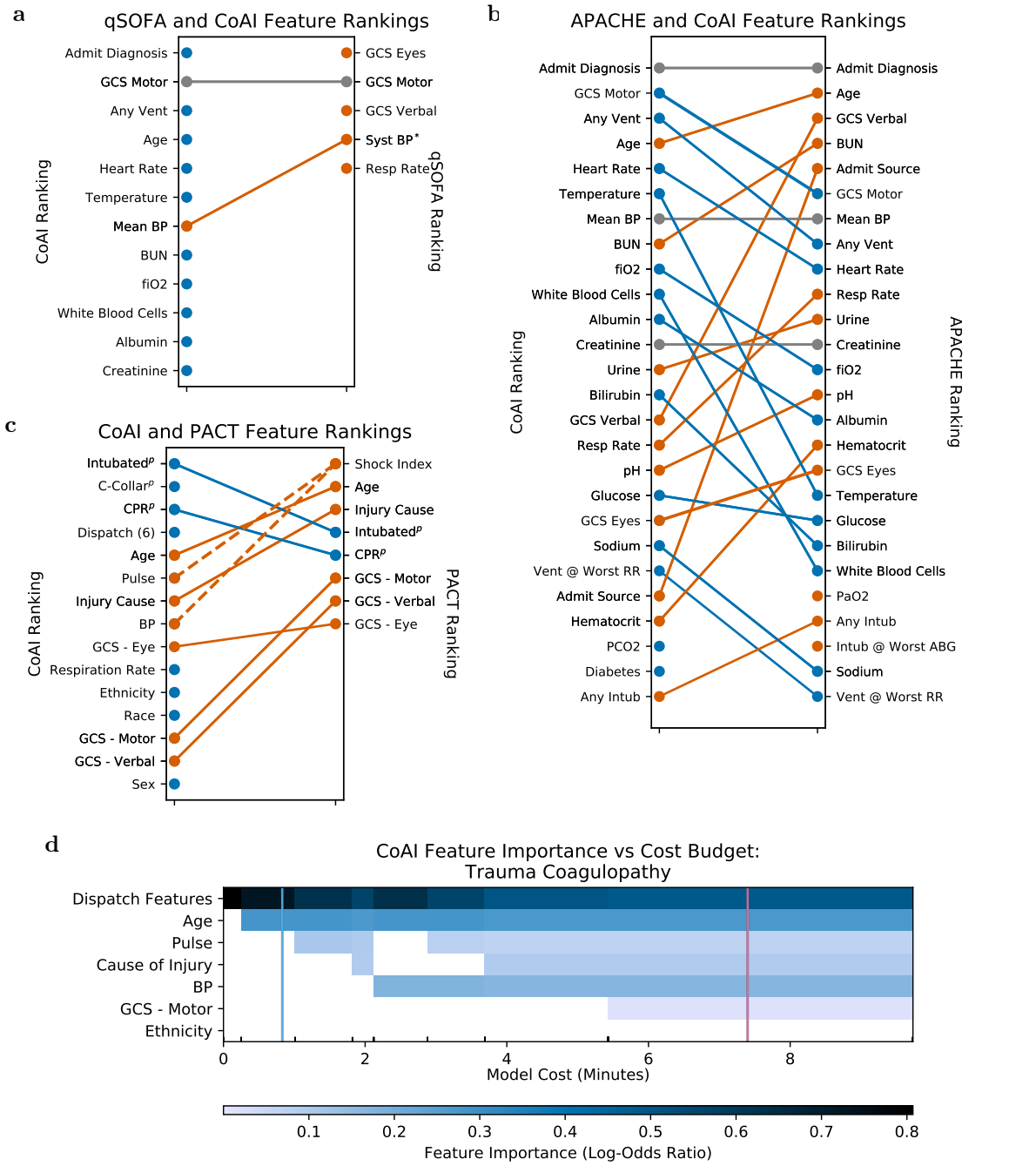


Figure 2.5: Importance of features selected by CoAI and other risk scores, ranked by order added to the model for CoAI and by regression coefficient or feature importance for others (see Section 2.2.4). Orange features are ranked lower in CoAI than the corresponding clinical risk score, and blue ranked higher. Gray indicates no change. CoAI is compared to (a) qSOFA and (b) APACHE for the ICU dataset. It is compared to (c) PACT for the trauma dataset. All features are shown for qSOFA and PACT. For CoAI, the full feature list is shown in (c), except for additional 0-importance dispatch features which are not shown. In (a) and (b) 12 and 25 CoAI features are shown, respectively, for clarity. Dashed line in (c) indicates that shock index is calculated from both pulse and BP. (d) Each heatmap column shows the importance ϕ_i of each feature for a CoAI model trained at a particular budget k in the trauma dataset. Dispatch and procedure features have zero cost and are grouped into one row. Darker blues indicate more importance. Columns have varying widths since they are scaled to align with each model's feature cost on the x-axis (in minutes). The left vertical blue line shows the EMS provider-preferred time budget; the right pink line shows the PACT time cost.

*Asterisk in a) indicates that systolic BP is measured simultaneously with, but is not identical to, mean BP. ^{P} in c) indicates a procedure variable.

2.3 Discussion

As AI and ML models become increasingly prevalent in healthcare, they risk imposing a large data-gathering burden on health care providers unless they can automatically select highly informative, easily acquired features. Our study is the first, to our knowledge, to survey clinical experts, build improved cost-aware clinical risk scores, and evaluate cost-aware models at operating points chosen by clinical providers.

Our framework, CoAI, is simple, flexible, and can efficiently adapt any predictive model to make cost-aware predictions. In the trauma dataset, CoAI's sophisticated AI models and its automated choice of low-cost, high-information features produce predictions more accurate than existing clinical scores at one-ninth the data-gathering cost. Analogously, for ICU mortality prediction, CoAI outperforms existing risk scores along several performance axes while using far fewer features. CoAI also makes accurate predictions at low monetary cost in an outpatient mortality prediction dataset, outperforming other AI methods for cost-aware prediction.

CoAI has several desirable properties that make it better suited to cost-aware clinical risk scoring than other AI methods. Its model-agnostic nature allows using complex models for large or nonlinear datasets and simple models for small datasets or ones where linear relationships are expected. Its ability to handle grouped features also makes it a natural fit for data with feature transforms or one-hot encoding and for data where properties of the data acquisition process, such as lab tests that return multiple measurements, result in naturally grouped features. By decoupling feature selection from model training, it can greatly improve robustness to cost shifts, adaptability to new target costs, and training time. Finally, CoAI always requests the same features for a given time budget and cost vector, which increases predictability for healthcare providers; methods like RL or decision trees may ask for different features for different patients.

There are several avenues for future work with CoAI. Like other low-cost AI methods, CoAI assumes the cost of a feature set is additive – equal to the sum of individual feature costs. This may be reasonable in ambulances or outpatient clinics, where a single provider must perform tasks sequentially, but the additivity assumption may need to be relaxed in other settings, like large hospitals, where multiple providers can perform different exams and tests simultaneously. CoAI also does not fully account for feature interactions in which including a particular feature changes the relative value of other features. While CoAI performed better in our experiments than approaches like CEGB and CWCF that explicitly account for feature interactions, a version of CoAI that accounts for interactions could improve performance even further.

Overall, we believe CoAI has demonstrated the potential to significantly improve clinical risk prediction. Its design considers the ease of gathering features to be just as important as the accuracy of predictions made using them. Our software is easy-to-use and integrated with existing open-source frameworks. We believe CoAI will make clinical risk scores more cost-aware, more accurate, and more effective at saving lives.

Acknowledgements

This work was funded by the National Science Foundation [CAREER DBI-1552309, and DBI-1759487], American Cancer Society [127332-RSG-15-097-01-TBG], and National Institutes of Health [F30 HL 151074, R35 GM 128638, and R01 NIA AG 061132]. Thanks to Scott Lundberg, Ian Covert, and the Lee Lab for helpful discussions about this paper.

2.4 Methods

2.4.1 Datasets

Trauma data. The trauma data used in this study were gathered over a 10-year period (2007 to 2017) and encompassed over 14,463 emergency department admissions for traumatic injury at a Level 1 Trauma Center. We selected 46 variables that were available in the pre-hospital setting, including dispatch information (injury date, time, cause, and location), demographic information (age, sex), and prehospital vital signs (blood pressure, heart rate, respiratory rate).

The outcome in this data was acute traumatic coagulopathy (ATC). We followed [24] and defined ATC as a binary outcome based on emergency department lab measurements of International Normalized Ratio (INR), where measurements greater than 1.5 were defined as coagulopathy.

ICU data. The ICU data were gathered from the public PhysioNet eICU repository [47]. The data come from over 142,139 patient admissions at 208 US hospitals between 2014 and 2015. While many features are available in this data, we selected 43 variables that had been preprocessed into a tabular format. Many of these variables are also used in the calculation of other risk scores, e.g., APACHE and APS. The outcome in this data was in-hospital mortality.

Outpatient data. The outpatient data were gathered from the NHANES I study, which is publicly available, with a reprocessed version for AI and ML recently released [29, 39]. The study gathered data on 13,442 individuals from 1971 to 1974, then followed up in 1992 to record 10-year mortality data. The NHANES data is unique among our datasets because: (1) it contains relatively healthy outpatients rather than relatively sick inpatients, (2) each row is an individual patient, as opposed to the trauma and outpatient dataset where stays are unique but patients may not be, and (3) it is a curated cross-sectional study rather than a convenience sample of patients who presented to the hospital, reducing dataset bias. We selected 35 features from this dataset including demographics, physical exam findings, and lab values. The outcome in this dataset was 10-year mortality.

Repeated measurements: In the trauma and ICU datasets, each sample is a patient visit, so a given patient who has visited the hospital multiple times could be represented as multiple samples in the dataset. In the outpatient dataset, each sample is a unique patient.

Data Processing. We standardized all variables to have 0 mean and unit variance. We mean-imputed missing data for input to all AI models except for GBMs, which naturally handle unprocessed missing data. Missing data handling in existing clinical risk scores is discussed further in Methods 2.4.9. We treated categorical variables as categorical in LightGBM [48]. For the trauma and outpatient datasets, we used a random 64/16/20 train/validation/test split. For the ICU dataset, patients were grouped into four geographic regions of the United States; we split off 1 region as a test set and split patients from the other 3 regions into train/validation sets using an 80/20 split by hospital. Labels were binary for all datasets, and we included only the patients from each dataset for whom label data were available.

2.4.2 Prehospital Time Costs: Survey Methodology

We gathered time costs for the trauma dataset by surveying professional prehospital care providers from the Pacific Northwest. We designed a Qualtrics survey to gather information on previous EMS experience, past experiences with and thoughts about computerized risk scores, costs for each feature in terms of objective time and subjective effort, and overall impressions of the value of risk scores in prehospital medicine. The full list of survey questions, and summary data on responses, is in Supplement Section 2.5.2. Free text answers are not included to preserve anonymity. We used anonymous links to send the surveys to all employees of 3 major emergency medical services and collected all results from September 19 to October 25, 2019 for analysis. Costs for each feature were determined by the mean cost across respondents. The survey did not include questions on time or effort cost for variables already present at dispatch, for which a cost of 0 was assigned. While we gathered data on how long each procedure in our dataset took to perform, we set all procedure feature costs to zero, since entering procedure information in the risk score requires simply knowing whether the procedure was done.

2.4.3 Outpatient Monetary Costs

We assigned costs to features in the outpatient data by referencing Medicare data on payments for lab tests [49]. Physical exam and other free measurements were assigned a cost of zero. Unavailable costs were mean-imputed. The full list of feature costs is in Supplementary Figure 2.7.

2.4.4 CoAI Method

CoAI is a feature importance-based method for cost-aware prediction. Given a model f trained on the full dataset and a single patient's features x_j, CoAI uses recently developed axiomatic feature importance methods to assign an importance ϕ_i to each feature i

CoAI uses these importance values to form an optimization problem that finds the best subset of features F – the one with the greatest sum of feature importances ϕ_i – that do not exceed the cost budget. Mathematically, this can be written $F = \arg\max_S \sum_{i \in S} \phi_i \quad \text{s.t.} \sum_{i \in S} c_i \leq k$, where c_i is the cost of feature i and k is the cost budget. We can solve this well-known optimization problem, known as a *knapsack problem*, with arbitrarily small error in polynomial time with respect to the number of features due to a fully polynomial time approximation scheme (we use the Google OR-tools solver) [35, 36].

For any AI/ML model, this approach lets us find the exact set of features that most reduce the model's loss within any cost budget under the assumption that features' effects are independent. If features interact, the attributions used are still the best mean-squared error approximation of the features' true effect on the loss averaged over all possible combinations of features that could be included or not included in the model. We can impute the missing features at test time, or train a new model on the selected feature subset. For consistency in the retraining case, the new model is of the same model class with the same hyperparameters as the all-features model used to calculate feature importance. We tested several other methods for using feature attributions and costs to select the best feature subset within a cost budget, including a greedy algorithm, recursive feature elimination, and knapsack method without retraining. These methods did not match the performance of our knapsack method with retraining but may prove useful in specific situations (Supplementary Figure 2.8 and Methods 2.4.6)

For interpretability purposes, we add small pseudocosts to zero-cost features (small enough that the sum of all such pseudocosts is less than the difference between any two non-zero-cost features). This is not strictly necessary, but does allow ranking of zero-cost features if desired.

2.4.5 Feature Attribution Methods

We needed measures of feature importance as well as feature cost to perform our CoAI analysis. We calculated feature importance using the SAGE global importance measure, which incorporates the SHAP (Shapley Additive Explanations) framework [40, 39, 38], in which a feature's importance is calculated with respect to a predictive model. The change of the model's loss when the feature is masked is recorded across all possible subsets of features, yielding an average change in prediction resulting from the inclusion of a feature in the model:

$$\phi_i = \frac{1}{m} \sum_{T \subseteq M \setminus \{i\}} \binom{m-1}{|T|}^{-1} \left(\mathbb{E}\left[\ell(y, f_T(X_T))\right] - \mathbb{E}\left[\ell(y, f_T(X_{T \cup \{i\}}))\right] \right) \tag{2.2}$$

where f is the model, M is the set containing all features, ℓ is the loss function (i.e., mean-squared error or cross-entropy), and $f_T(X_T)$ represents the expectation of the model's output when only the values of feature set T in the dataset are known (the other features are masked and marginalized over).

SAGE estimates of feature importance can be calculated for any AI/ML model and run quickly in our experiments. For the GBM, MLP, and linear models presented here, it is also possible to make use of fast algorithms for calculating SHAP values in decision trees and linear models, as well as fast calculation of the related Aumann-Shapley values for deep models [40, 39, 50]. While we do not do this in the paper, it would improves runtime and reduces variability in importance estimates, at the cost of, in some cases, losing the valuable theoretical properties of SAGE. It is also worth noting that CoAI is compatible with *any* feature attribution method that assigns a measure of importance ϕ_i to each feature – SAGE and SHAP are not the only methods of this type, but are model-agnostic and satisfy desirable axioms for the use case we consider here.

2.4.6 Alternate Optimization Methods

We tested three additional methods to search for feature sets with high total feature importance and low measurement cost:

1. **Greedy selection.** In the greedy selection version of the algorithm, features are sorted by their importance divided by cost: $\omega_i = \frac{\phi_i}{c_i}$. Features are then added to the model one by one, from highest to lowest value of ω_i, until no more can be added without exceeding the cost budget. This very simple method works reasonably well.

2. **Recursive feature elimination.** This is inspired by the recursive feature elimination method often used for feature selection in linear models. A model is trained on the full dataset with all features. A measure of feature quality is calculated, and the lowest-quality feature is removed. Another model is trained on the resulting dataset, and the process iterates until only the desired number of features are left. For cost-aware prediction, the quality measure is simply importance divided by cost: $\omega_i = \frac{\phi_i}{c_i}$, and the iteration stops when model cost is below the budget constraint. This method is valuable for its ties to existing feature selection literature. Because the process is stepwise and feature importance can be recalculated at each iteration, the algorithm can account for feature dependence, where removing one feature changes the importance of other features. Though it performs well it, does not outperform our knapsack method.

3. **Knapsack without retraining.** This is a simple precursor to Impute-CoAI; after a feature subset is selected, those features are fed into the original model that was trained on all variables. Other features are mean-imputed. This greatly increases training speed but decreases performance, a phenomenon observed in previous work [42]. This motivates both NM-CoAI and I-CoAI; we must do more than simple mean-imputation of missing features, whether that is training a new model on the restricted feature set or accurately imputing the missing features.

Performance plots of these alternate methods on a random train-test split of the trauma dataset are shown in Supplemental Figure 2.8.

2.4.7 Base Model Training Details

We used three classes of base models: gradient boosting machines, logistic regression, and feedforward neural network or multi-layer perceptron (MLP) [**friedman2001elements**, 51, 52]. Hyperparameters were selected using the train/validation splits described in Methods 2.4.1 with all features then fixed for cost-aware learning. The train and validation sets were combined for training after parameters were fixed. We implemented gradient boosting machines using the LightGBM package [48] and used the following parameters:

- Learning rate: 0.01
- Maximum Number of Trees: 1000
- Early Stopping Rounds: 100
- Max Tree Depth: $\in \{1, 2, 4, 8, 10\}$
- Gamma: 1.0
- Minimum child weight: 10
- Subsampling: $\in \{0.2, 0.5, 0.8, 1.0\}$

We implemented logistic regression models with Scikit-learn [15] and used the following parameters:

- Regularization type: $\in \{\text{L1}, \text{L2}\}$
- Regularization strength: $\in \{10^{-5}, 10^{-4}...10^{4}, 10^{5}\}$
- Solver: SAGA

We implemented MLPs using TensorFlow and Keras[53, 54], running for 10 epochs with the following parameters:

- Layers: $\in \{[16], [32], [64], [128], [256], [512], [64, 16], [128, 32], [256, 64], [256, 64, 16], [512, 128, 32]\}$
- Dropout probability: $\in \{0., 0.25, .5, .75\}$

Parameter values not specified above were left at their default values.

2.4.8 Previous Clinical Risk Scores

Several scores have been developed for the specific case of ATC in trauma patients. The COAST score – an additive point-based score using abdominal/pelvic injury, chest decompression, temperature, systolic blood pressure, and entrapment – was one of the earliest [25]. The subsequent PACT score was a six-feature logistic regression involving shock index, age, mechanism of injury, Glasgow Coma Score, and prehospital CPR and intubation [24]. Both models use relatively simple prediction methods and a fixed set of features that limit the range of time budgets in which they can be used. A recent study developed a linear model to predict whether military trauma patients would receive massive transfusion of blood products, with the goal of discovering "concrete and rapidly and easily assessable" predictors. This study did not explicitly account for model cost, but noted that some data, including vital signs, may be difficult to acquire or unavailable in the prehospital setting and developed multiple models with different numbers of features, implying the potential value in this area of models that automatically account for cost [46].

Many risk scores exist to predict mortality of ICU patients. The most popular include the APACHE, APS, SOFA, and qSOFA models [27, 28]. Most of these models take as input a large number of features, while the qSOFA score uses only the Glasgow Coma Score, respiratory rate, and blood pressure at the cost of worse predictive performance. These risk scores all use linear or additive models that aim to either achieve high accuracy with many features, or moderate accuracy with few features. Although mortality prediction in critically ill patients is a topic of great interest in medicine, only a small number of feature sets have been explored. There is no single published model that can make accurate predictions within any feature budget. Finally, although outpatient survival prediction is an important task, we are not aware of a standard clinical tool for this purpose.

2.4.9 Implementation of Existing Clinical Models

We compared to the following clinical models: qSOFA, APS, APACHE IIIa, and APACHE IVa in the ICU dataset, and the PACT score in the trauma dataset. APS, APACHE IIIa, and APACHE IVa were pre-calculated for the ICU dataset. We re-implemented the qSOFA and PACT scores by referring to their respective publications [28, 24]. Notably, the qSOFA score required systolic rather than mean blood pressure as an input variable, so we extracted systolic blood pressure data from the eICU dataset and only gave qSOFA access to this variable. We handled missing data in qSOFA by assuming the corresponding binary variable was false (i.e., the input "respiratory rate greater than 22" was always false if respiratory rate was missing). We handled missing data in PACT with mean imputation. We also found that re-training logistic regression models on the variables in the PACT model substantially improved performance. The final plots show results from the re-trained PACT regression.

In Figure 2.5, we ranked features in each clinical risk score by their importance. We ranked PACT features by standardized regression coefficient. The qSOFA score assigned equal weight to all features, so the ordering was arbitrary. The APACHE IVa score did not publish standardized regression coefficients, so we trained a model to mimic the APACHE IVa score in our dataset, which achieved a held-out Spearman R^2 of 0.98 on validation data. We then calculated feature importance for this model using LightGBM's implementation of Shapley value feature importance for trees.

2.4.10 Other Cost-aware prediction methods

Cost-aware prediction is a topic of growing interest in ML and AI. Established techniques, like the LASSO penalty in regression, encourage models to rely on few of their input features but do not generally incorporate the idea that different features may have different costs [55]. More recent methods have attempted to minimize

the feature acquisition cost for each individual prediction while maximizing its accuracy. Methods involve either perturbing an existing model to determine the most important features [56], using decision trees to divide the data into similar groups while penalizing splits that use expensive features [57, 58, 16], or applying reinforcement learning (RL) approaches, which use deep learning to simulate the process of asking for features one at a time and then making a prediction [17, 59, 60, 61]. In this paper, we estimate feature importance using state-of-the art axiomatic methods that guarantee features with a greater effect on the output will be ranked more highly. This can be seen as an improvement on perturbation-based methods and allows CoAI to accurately choose the most important features within a given cost budget [40, 39].

Despite the emergence of methods for low-cost AI, scant research has used these methods to produce risk scores for real clinical problems. Many approaches are evaluated on toy datasets with random or arbitrary feature costs. We know of only one paper that evaluated cost-aware methods on a clinical prediction task, which used a Mechanical Turk survey to gather costs from laypeople rather than attempting to synthesize expert opinion [61]. While study was a valuable attempt to reduce the burden of diagnosis for patients, because costs were measured on a 1-10 subjective scale of convenience it is not clear how to add costs together or interpret the resulting total model cost. This results in a model with unclear implications for clinical practice. On the contrary, our work uses costs gathered from expert clinicians in units of minutes or dollars, where total model cost has clear clinical implications.

2.4.11 CEGB Implementation Details

We implemented Cost-Effective Gradient Boosting using the authors' code which is integrated into LightGBM [16]. We used the `cegb_penalty_feature_coupled` parameter to pass the per-feature cost vector and `cegb_tradeoff` to control the cost-performance tradeoff. The `cegb_penalty_feature_coupled` parameter charges a global cost the first time a feature is used in any tree. While other options exist, such as charging a cost the first time a feature is used for each sample (resulting in different features being measured for different samples), the coupled penalty is the most apt comparison to CoAI, since it selects a low-cost set of features to be measured for all patients. We tuned the `cegb_tradeoff` parameter on 101 logarithmically spaced points in $[10^{-5}, 10^{5}]$.

2.4.12 RL Implementation Details

We implemented the reinforcement learning approach of [17] using the authors' published code. We used hyperparameter values for most parameters from the example code with a dataset closest in size to ours (the Miniboone dataset). We set the neural network's hidden layer size to 128 and the "difficulty" (a multiplier on the number of steps for training, early stopping, etc) to 1000. We tuned the regularization strength with 16 logarithmically spaced points: $\{10^{-14}, 10^{-13}...10^{1}\}$. We trained two models at each regularization value because we noticed substantial variation in performance between runs with identical parameters. This resulted in fewer points on the cost-performance curve than CoAI or CEGB but was necessary due to the method's slow runtime.

The reinforcement learning approach was not directly comparable to CoAI because it is allowed to choose different features for each patient. Thus, it may attain a lower cost that is unattainable by a model with a single fixed list of features. However, we could not easily alter this property so we did not change this behavior. In all figures, we use the "average-budget" version of CWCF to provide a generous upper bound on performance, though we note that CWCF is capable of using a hard budget like CoAI at the cost of lower predictive power.

Reinforcement learning suffered in our tests because it makes dichotomous predictions by default, which dramatically reduces its performance in a ranking-based metric like AU-ROC. We attempted to account for this by editing the code to dump the final Q values for each sample at test time. Because Q-values correspond to expected reward for taking an action, we interpret the Q-values for the "classify into a given class" action as a measure of the model's confidence in predicting that class. Using Q-values as pseudo-probabilities improves classification metrics like AU-ROC but still underperforms, perhaps in part because Q-values in an RL task do not lead to as well-calibrated probabilities as the outputs of a true classification model with a logistic objective.

2.4.13 Cost-Performance Curve Comparison

We compared several methods for low-cost AI modeling in Figure 2.3. We retrained each method on repeated random train-test splits of the dataset (50 splits for trauma, 9 for ICU and 100 for outpatient). Parameters were tuned separately on each train/validation/test split. Because CEGB and CWCF may produce models of different costs on each run, we calculated the mean and standard deviation of each model's cost-performance curves by interpolating them along 100 points, linearly spaced between the lowest and highest-cost models over the 100 runs. We used previous-value interpolation so each point was a conservative estimate of performance (i.e., if no 5-minute model existed for a run, the 4-minute model's performance would be used to interpolate performance at a 5-minute budget).

To calculate statistical significance, we used a two-sided paired-samples T-test, paired by random train-test split, on the mean AU-ROC for each model's cost-performance curves. The T-test normality assumption is justified by the central limit theorem and the fact that each sample in this test is a *mean* AU-ROC. Because CWCF ran slowly, we ran fewer replicates (5 for trauma and ICU, 11 for outpatient) and compared it with other methods using only the splits for which CWCF had a run completed. Tables of the resulting p-values are shown in Supplementary Figures 2.14-2.16.

In the outpatient data, we had to account for the fact that neither CEGB nor CWCF was able to use the information that features came in groups. We post-hoc adjusted CEGB's model costs by examining the features used by each CEGB model and re-assigning costs based on which *groups* were used. We were not able to post-hoc adjust CWCF's model costs, because the CWCF package is not aware of feature groups and reports only the model's total cost, not which features were used. Thus, we calculated an upper bound for CWCF by setting the performance of *all* CWCF models on a given train-test split to the maximum performance of *any* CWCF model on that split. This will never underestimate performance of CWCF at any budget, and will overestimate performance at all but one point.

2.4.14 Grouped Feature Costs

We extended CoAI to handle grouped feature costs, as encountered in the outpatient dataset, by performing the same knapsack optimization but over groups rather than features. Each group g has a single cost c_g for acquiring all features in that group. It also has an importance equal to the sum of importances for each feature in the group: $\phi_g = \sum_{i \in g} \phi_i$. We solve the knapsack problem:

$$F' = \arg\max_S \sum_{g \in S} \phi_g \quad \text{s.t.} \sum_{g \in S} c_g \leq k$$

This gives us a set F' of groups $g \in F'$. To identify the features for model training, we simply take the union of all features i in all groups in F': $F = \bigcup_g \{i | i \in G\}$. The model cost is the sum of group costs: $\sum_{g \in F'} c_g$.

2.4.15 Binary Search/Tuning Details

We compared the efficiency of model tuning by trying to build models under a particular cost budget with CoAI, CEGB, and reinforcement learning. This was straightforward with CoAI: we entered the desired cost k as a model parameter. CoAI trained one full-data model to calculate the ϕ_i values, selected maximum-importance features within that budget, and then trained the final model (two total trainings). It is a less straightforward process in other frameworks to determine how to satisfy a particular cost constraint while guaranteeing that the best possible model within the framework has been found.

In CEGB and reinforcement learning, we had to use blind tuning to achieve the same goal because the relationship between their cost-performance tradeoff parameter λ and the actual model cost is unclear and dependent on many factors (magnitude of the loss, scale of the costs, etc). All we know is that as λ increases, cost and accuracy should monotonically decrease. Thus, we performed binary search by setting upper and lower bounds on λ and a starting value λ_0. For rounds i from 0 to T, we iteratively train a model with λ_i and check if the model's cost is below our target k. If so, we set $\lambda_{i+1} = \frac{\lambda_i - \lambda_{\min}}{2}$. If not, we set $\lambda_{i+1} = \frac{\lambda_{\max} - \lambda_i}{2}$. We then continue the iteration.

Values used for this search are:

- $\lambda_{\min} = 0$

- $\lambda_{\max} = 10^6$
- $\lambda_0 = 1$
- $T = 32$

Data Availability

Two of our three datasets – the ICU and outpatient datasets – are publicly available. The ICU dataset was published in [47] and is available from the MIT eICU Collaborative Research Database (`https://eicu-crd.mit.edu/gettingstarted/overview/`) but requires approval before download. The outpatient dataset is a subset of the NHANES I study [29] and was published in its current format in [39]. It is also uploaded to our Github repository along with our code (see below). The trauma dataset is not publicly available due to patient privacy concerns.

Code Availability

Code implementing CoAI is available at `https://github.com/suinleelab/coai`. The repository also includes notebooks reproducing the results that do not rely on the trauma dataset, including performance and feature importance for CoAI and existing mortality risk scores on the ICU dataset and comparisons with existing low-cost AI methods on the outpatient dataset.

Institutional review board statement

The survey data for this study was gathered under an exempt determination from the University of Washington Institutional Review Board (Human Subjects Division, STUDY00006890).

2.5 Supplementary Material

2.5.1 Supplementary Figures

	Measurement Cost (Minutes)
Age	0.25
Agency Level from Scene	0.00
Agency Mode from Scene	0.00
Age (Units)	0.00
Cause of Injury	0.81
Ethnicity	0.55
Form from Scene	0.00
Race	0.46
Residence State	0.00
Destination Reason from Scene	0.00
First Blood Pressure on Scene	1.87
First Pulse on Scene	0.76
First Respiration Rate on Scene	1.03
GCS on Scene (Eyes)	1.04
GCS on Scene (Motor)	1.74
GCS on Scene (Verbal)	0.93
Assisted Respirations on Scene	0.00
Sex	0.29
Arrival Date to Scene (Month)	0.00
Arrival Date to Scene (Day)	0.00
Arrival Date to Scene (Weekday)	0.00
Departure Date (Month)	0.00
Departure Date (Day)	0.00
Departure Date (Weekday)	0.00
Injury Date (Month)	0.00
Injury Date (Day)	0.00
Injury Date (Weekday)	0.00
Notification Date to Scene (Month)	0.00
Notification Date to Scene (Day)	0.00
Notification Date to Scene (Weekday)	0.00
Arrival Time to Scene	0.00
Departure Time from Scene	0.00
Injury Time	0.00
Notification Time to Scene	0.00
Injury ZIP Code (km N of hospital)	0.00
Injury ZIP Code (km E of hospital)	0.00
Residence ZIP Code (km N of hospital)	0.00
Residence ZIP Code (km E of hospital)	0.00
Intubation (Procedure)	0.00
Other Splinting/Immobilization (Procedure)	0.00
IV Placement (Procedure)	0.00
Cervical Collar (Procedure)	0.00
Backboard (Procedure)	0.00
Supplemental Oxygen (Procedure)	0.00
Pelvic Binder/Sheeting (Procedure)	0.00
CPR (Procedure)	0.00

Figure 2.6: Survey Results: Estimated costs for all trauma features

	Feature Group Cost (Dollars)
BUN	4.39
Age	0.00
Alkaline Phosphatase	9.56
CBC w/Diff	8.63
Calcium	5.73
Cholesterol	14.88
Creatinine	5.69
Height	0.00
Hemoglobin	2.63
Physical Activity	0.00
CBC Auto	7.18
Potassium	5.11
Pulse Pressure	0.00
Red Blood Cells	3.35
Sedimentation Rate	3.00
Serum Albumin	5.50
Serum Protein	4.07
Sex	0.00
Sodium	5.35
Systolic BP	0.00
Total BIlirubin	5.57
Uric Acid	5.02
Urine Albumin	6.42
Urine Glucose	4.37
Urinalysis	2.41
Weight	0.00
SGOT	4.83

Figure 2.7: Numeric costs for outpatient features by group.

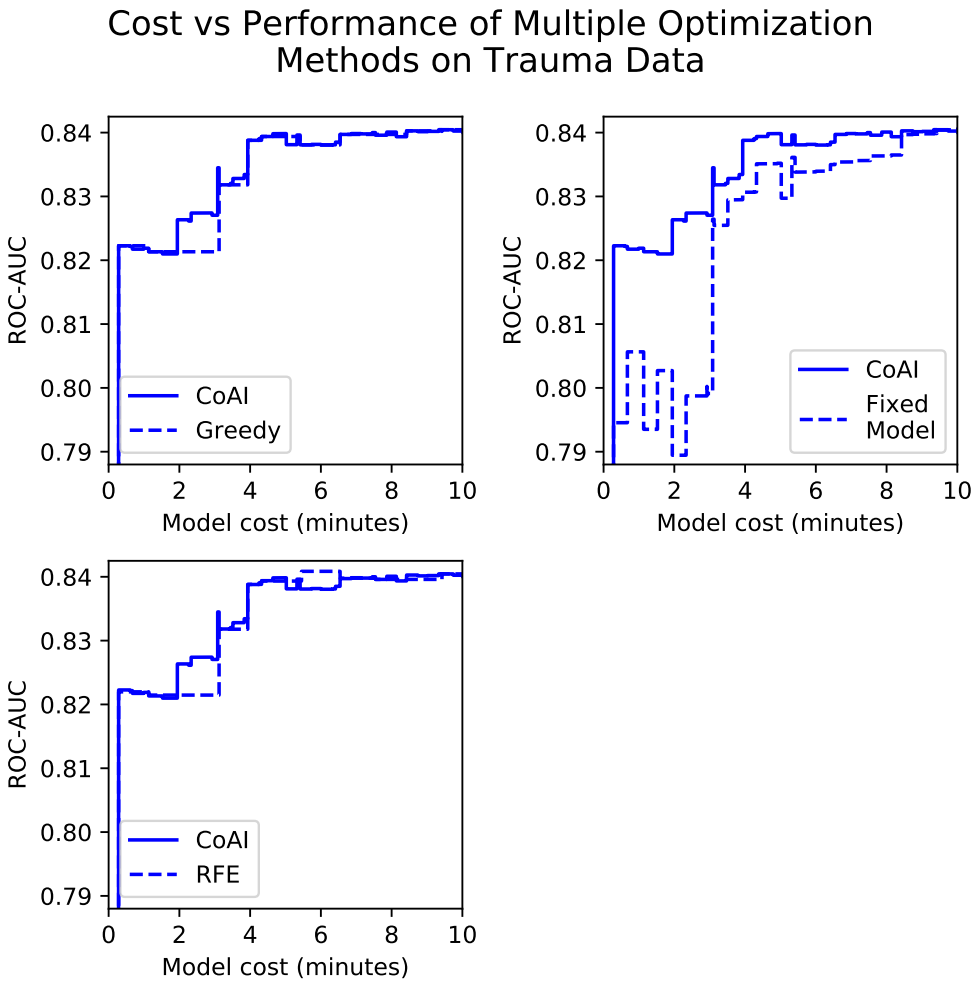


Figure 2.8: Performance of CoAI variants on a single train-test split of the trauma data, including a greedy solution to the knapsack problem where features are added in order of decreasing (importance divided by cost), a method using the same knapsack solver as the maintext but without model retraining, and a recursive feature elimination based method where the feature with lowest (importance divided by cost) is removed from the model, the model is retrained, and the process is repeated until the budget k is satisfied.

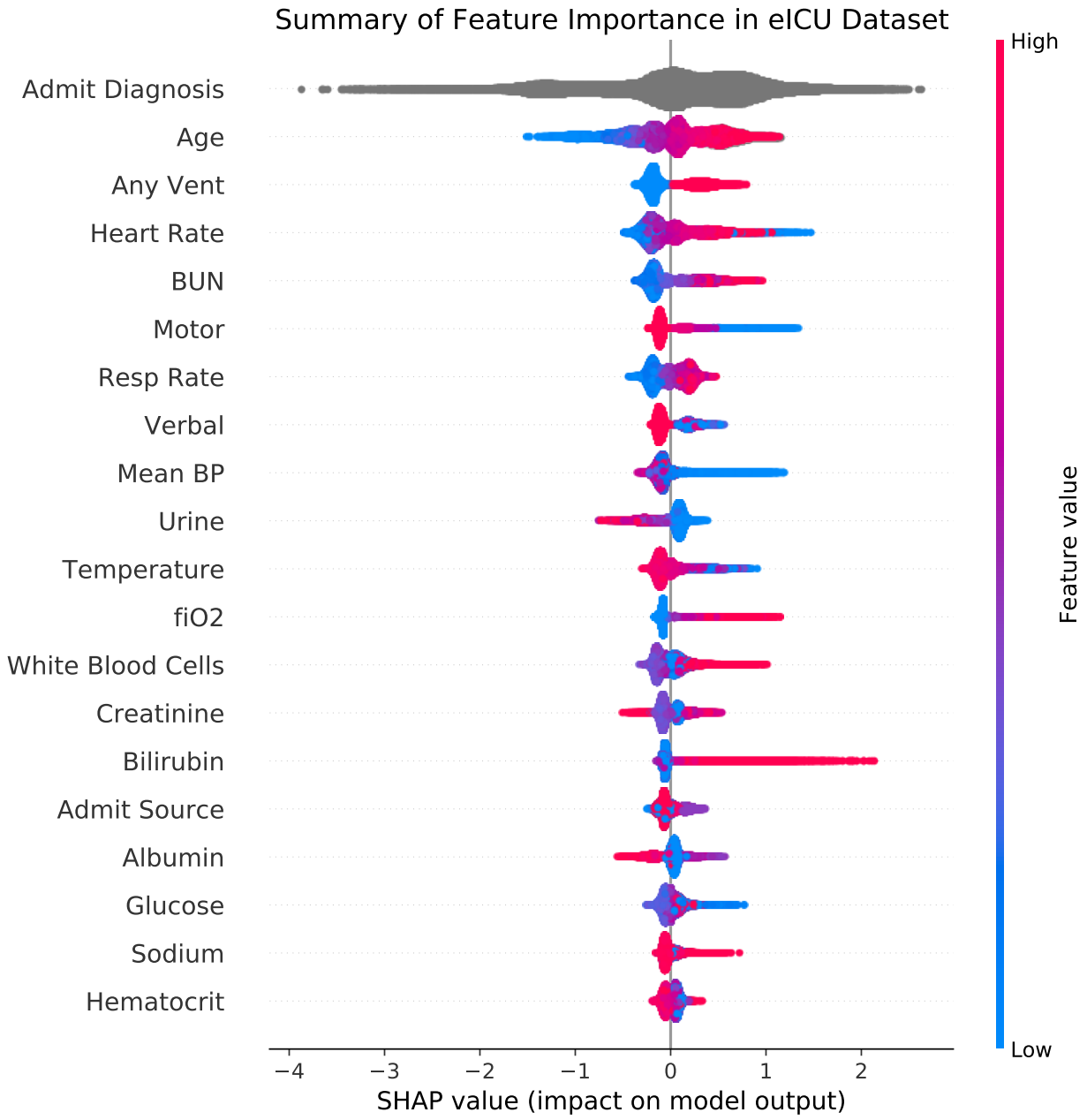


Figure 2.9: Summary of feature importance for predicting mortality from a GBM model trained on a random split of the eICU dataset.

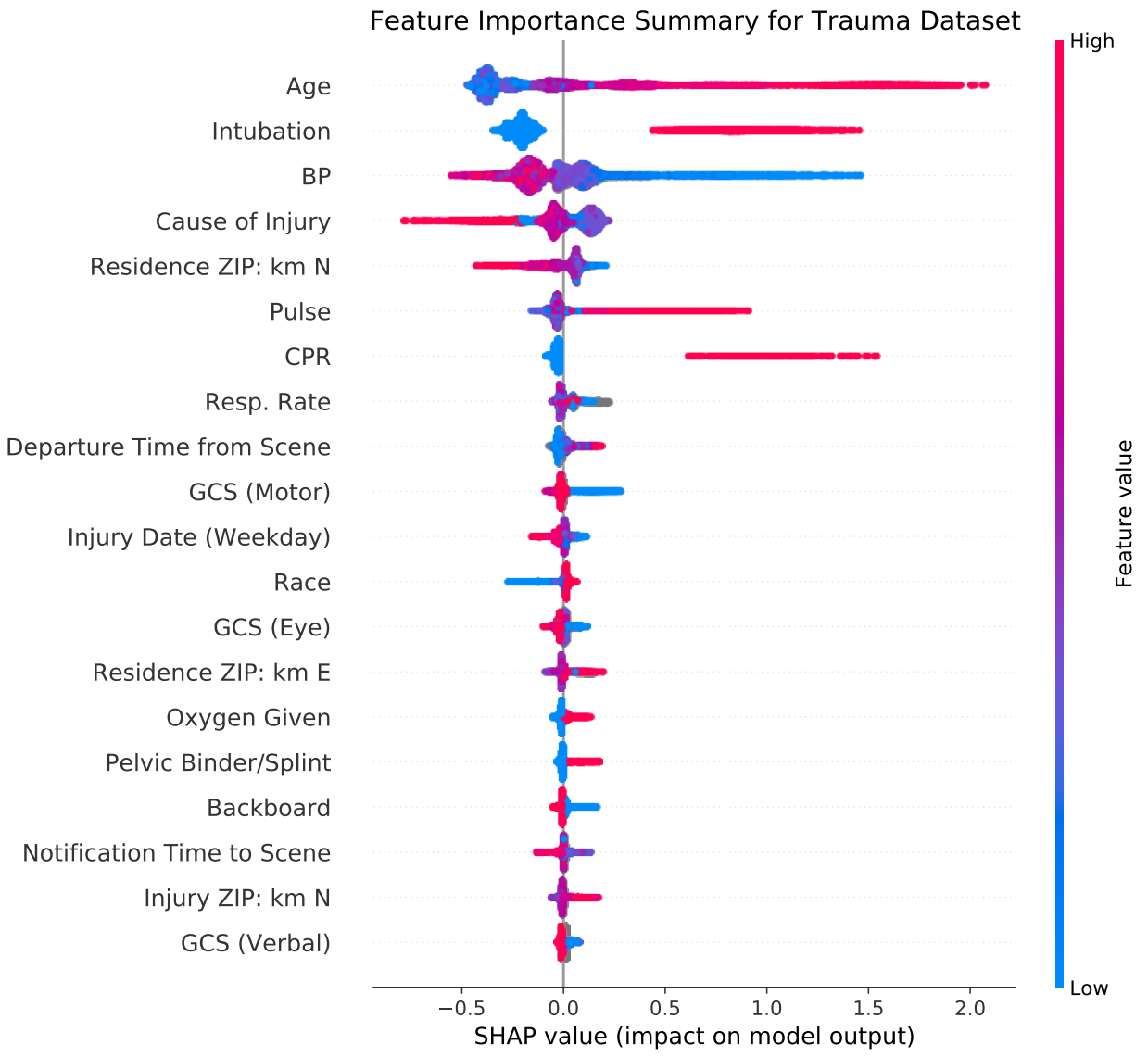


Figure 2.10: Summary of feature importance for predicting mortality from a GBM model trained on a random split of the trauma dataset.

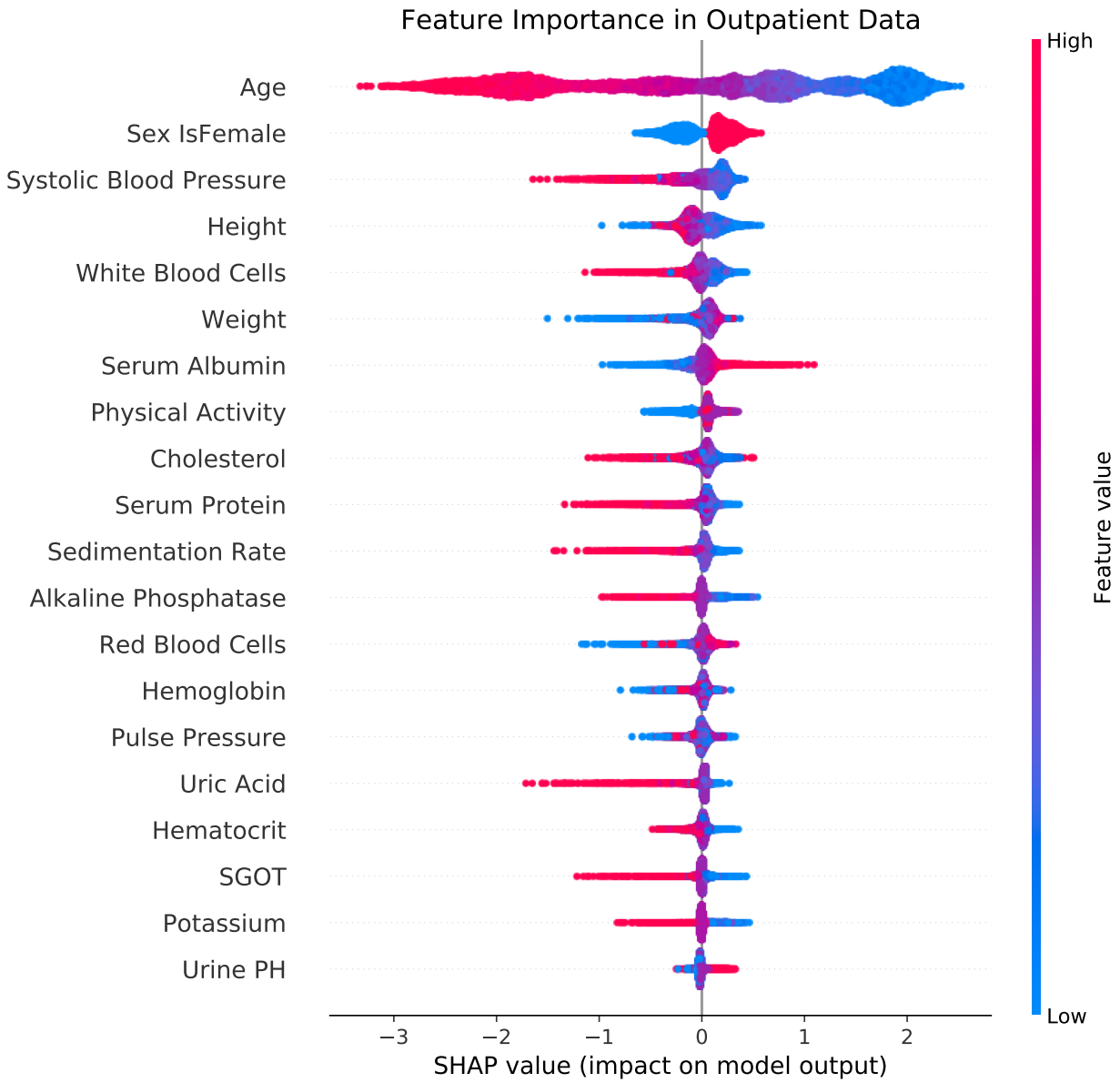


Figure 2.11: Summary of feature importance for predicting mortality from a GBM model trained on a random split of the outpatient dataset.

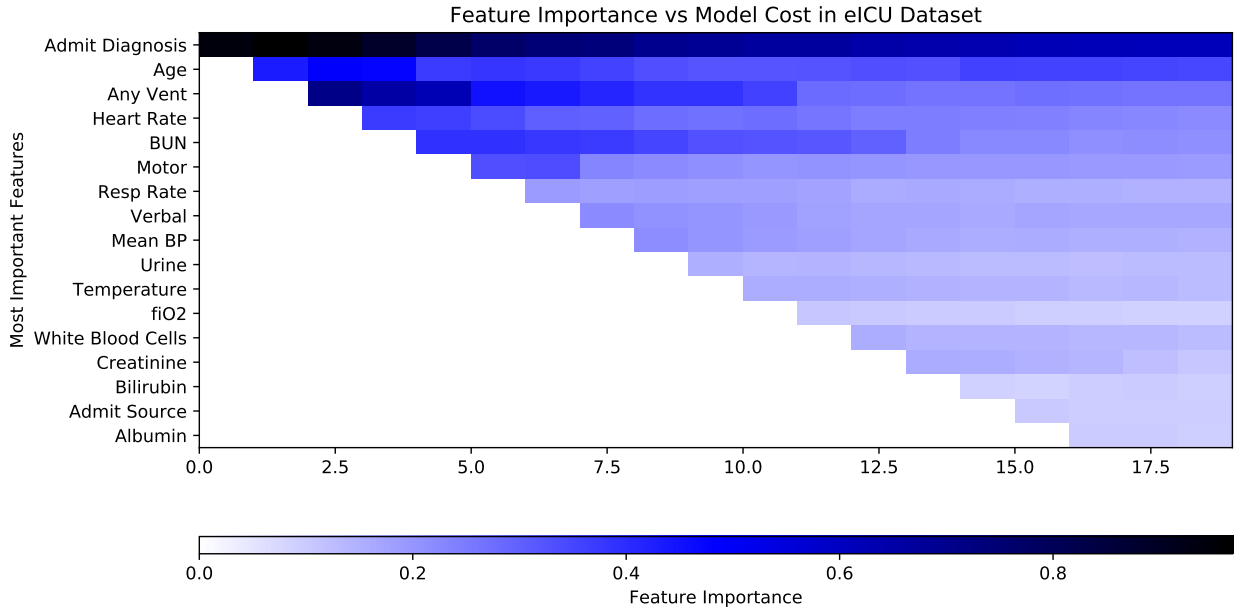


Figure 2.12: Feature importance heatmap for CoAI on a random train-test split of the eICU dataset.

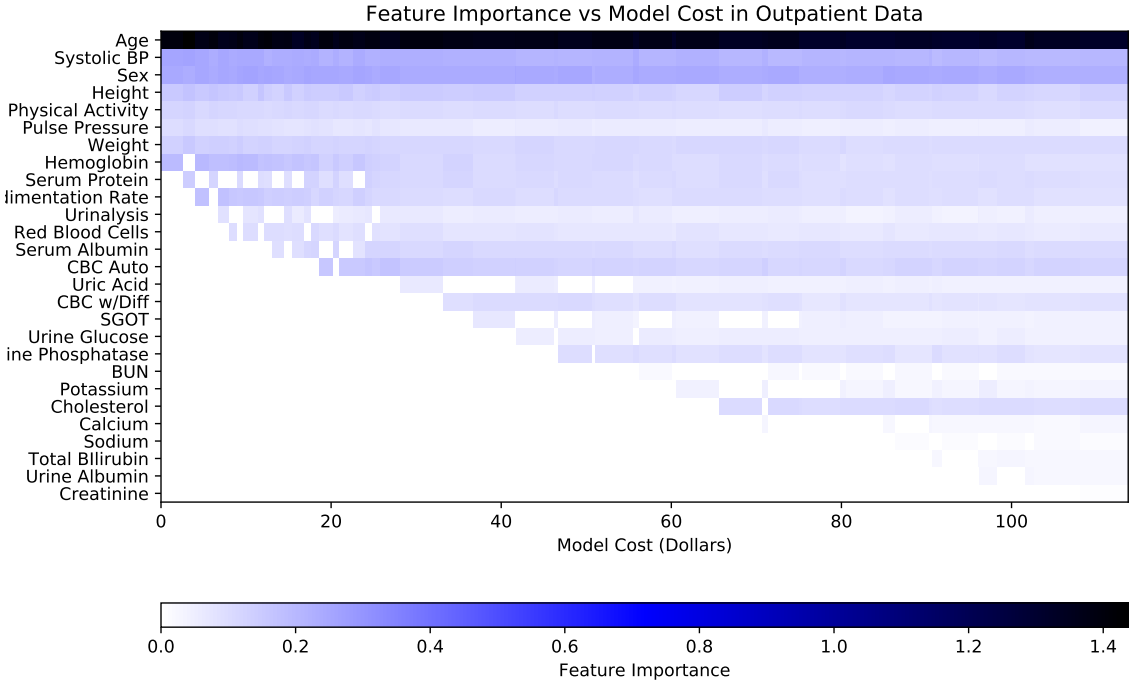


Figure 2.13: Feature importance heatmap for CoAI on a random train-test split of the outpatient dataset.

Method 1	Method 2	T-statistic	p-value
CoAI (GBM)	CoAI (Linear)	17.857775	4.303139e-23
CoAI (GBM)	CoAI (MLP)	19.099964	2.393716e-24
CoAI (GBM)	CEGB	7.494643	1.140670e-09
CoAI (GBM)	CWCF	12.688831	2.221739e-04

Figure 2.14: Statistical significance of performance differences between all methods on trauma dataset.

Method 1	Method 2	T-statistic	p-value
CoAI (GBM)	CoAI (Linear)	27.131571	3.668743e-09
CoAI (GBM)	CoAI (MLP)	8.167426	3.760826e-05
CoAI (GBM)	CEGB	12.409248	1.659725e-06
CoAI (GBM)	CWCF	15.029808	1.141900e-04

Figure 2.15: Statistical significance of performance differences between all methods on ICU dataset.

Method 1	Method 2	T-statistic	p-value
CoAI (GBM)	CoAI (Linear)	1.072023	2.863159e-01
CoAI (GBM)	CoAI (MLP)	23.848603	7.988987e-43
CoAI (GBM)	CEGB	11.241615	2.190554e-19
CoAI (GBM)	CWCF	3.524580	5.496269e-03

Figure 2.16: Statistical significance of performance differences between all methods on outpatient dataset.

2.5.2 EMS Provider Survey

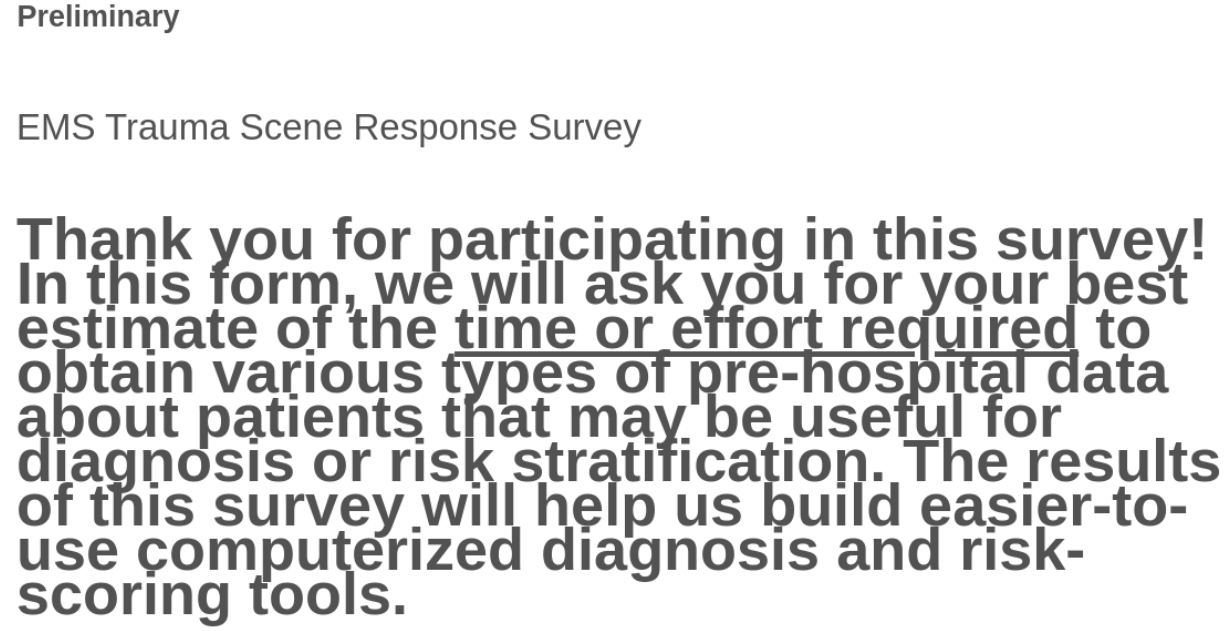


Figure 2.17: Survey form for trauma

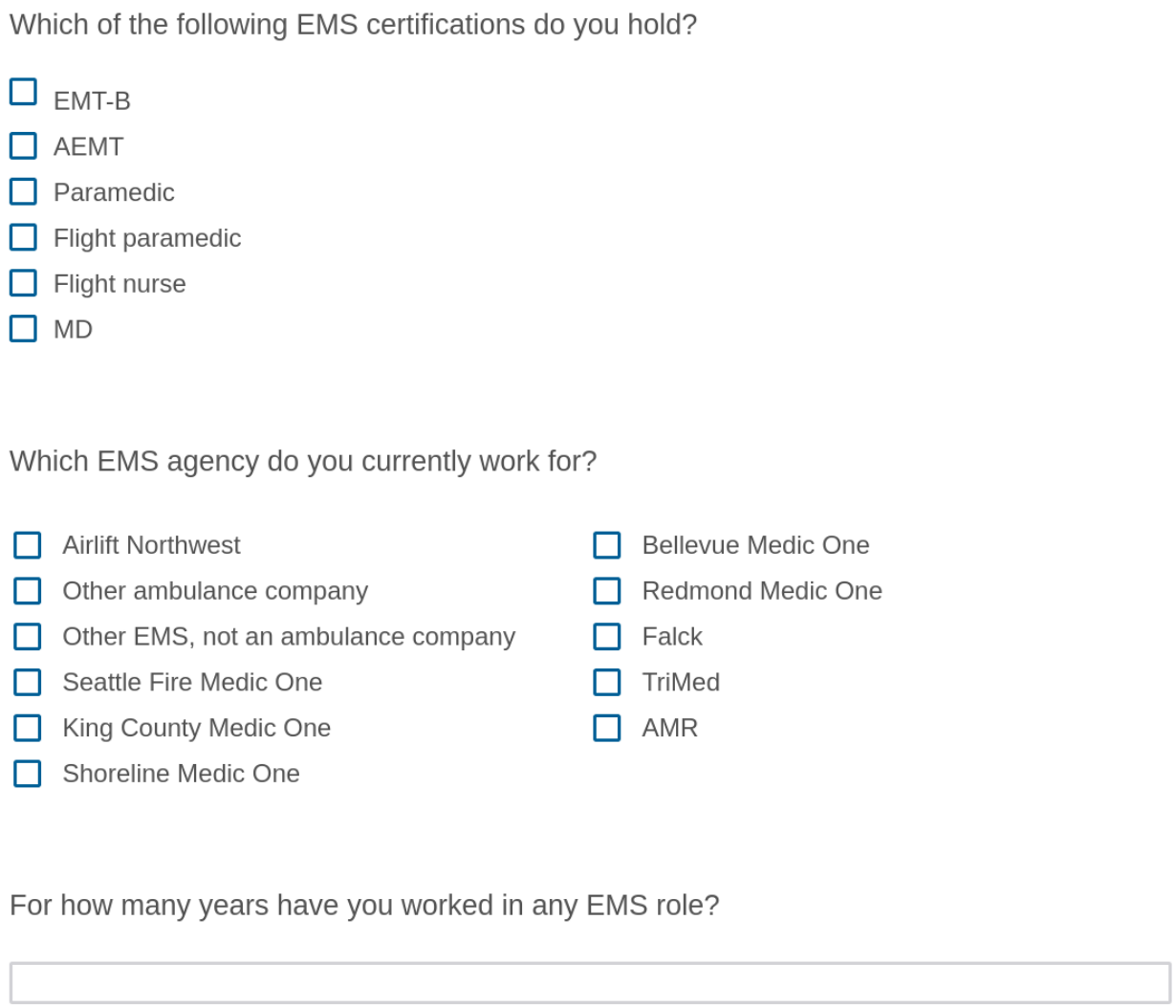


Figure 2.18: Survey form for trauma

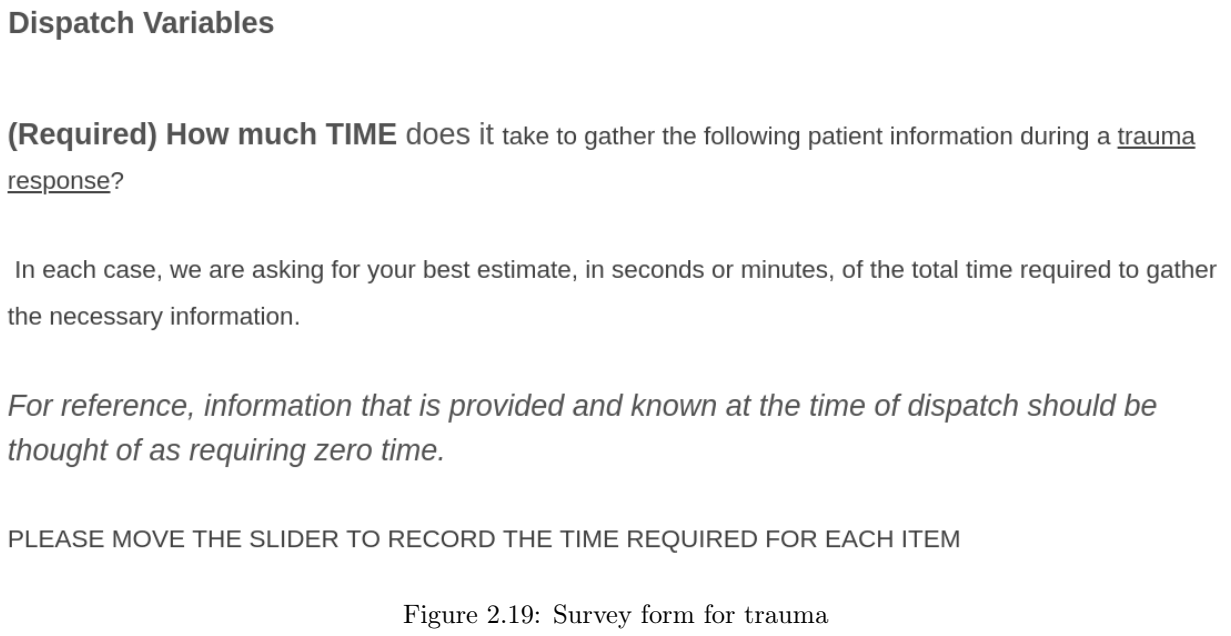


Figure 2.19: Survey form for trauma

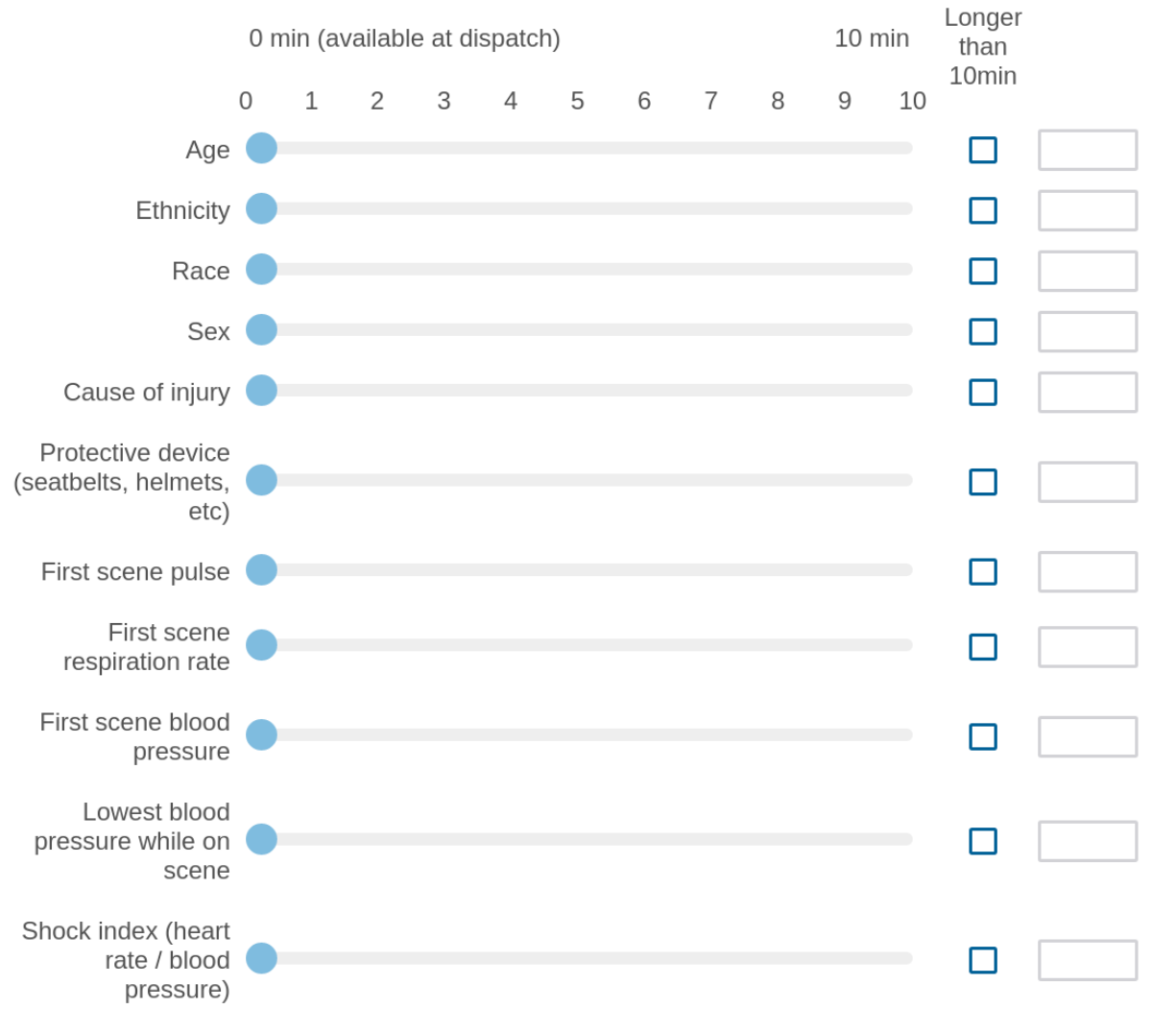


Figure 2.20: Survey form for trauma

Figure 2.21: Survey form for trauma

(Required) HOW DIFFICULT or burdensome it is to gather the following patient information during a trauma response?

We are asking for your best estimate, as a subjective rating of the amount of effort required, with 1 being the least and 10 being the most, to gather the necessary information.

For reference, a rating of 1 might be assigned to information that you have been provided at the time of dispatch and requires almost no effort. A rating of 10 might represent information that is very difficult to obtain, due to the need for detailed patient interview or close physical examination or an advanced procedural task.

PLEASE MOVE THE SLIDER TO RECORD THE LEVEL OF DIFFICULTY FOR EACH ITEM

Figure 2.22: Survey form for trauma

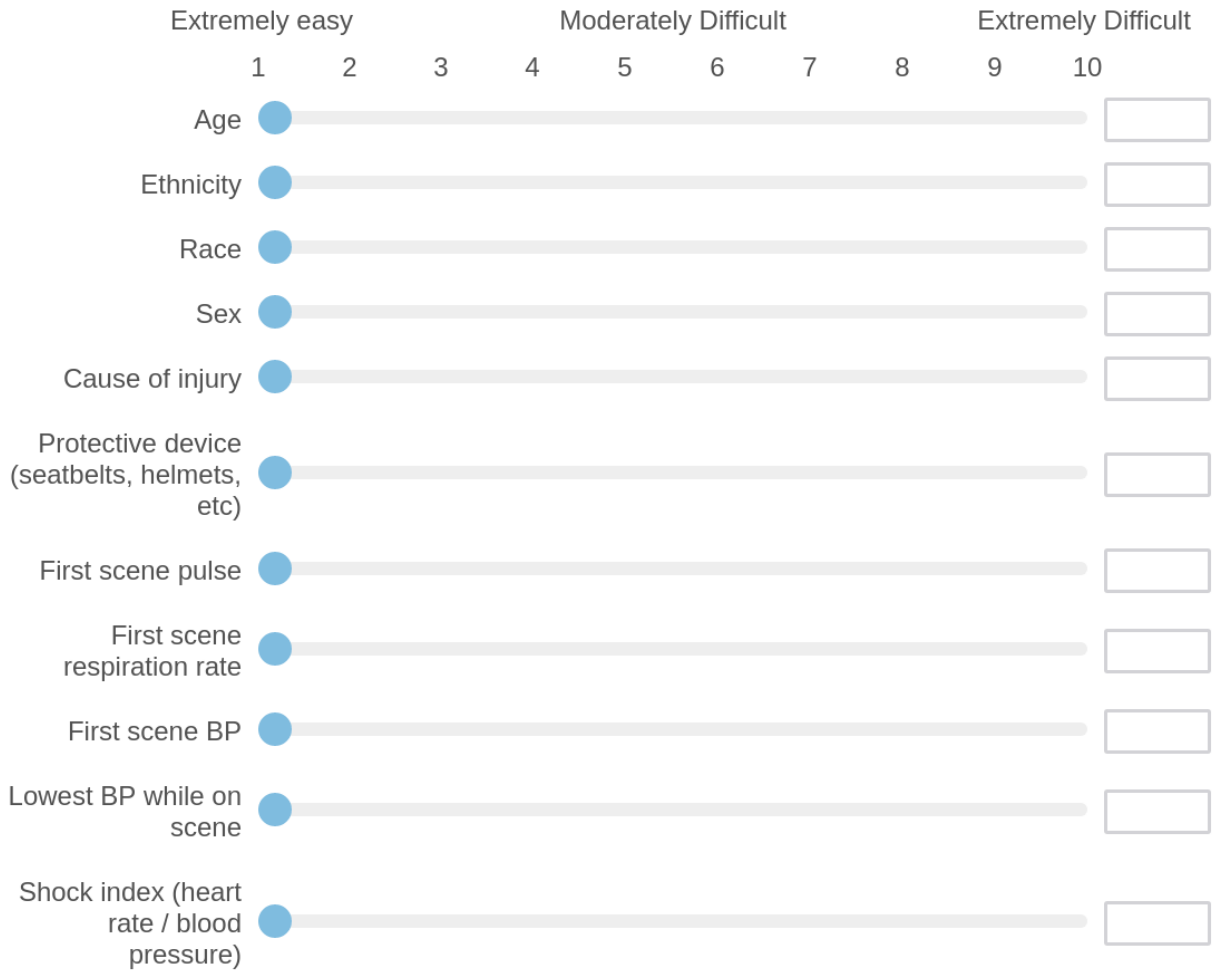


Figure 2.23: Survey form for trauma

Figure 2.24: Survey form for trauma

Block 3

The following variables are ***procedures*** rather than variables to be measured. For each procedure, please provide your best estimate, in seconds or minutes, of the total time required to **recognize that the procedure is necessary and to perform it**.

PLEASE MOVE THE SLIDER TO RECORD THE TIME REQUIRED FOR EACH ITEM

Figure 2.25: Survey form for trauma

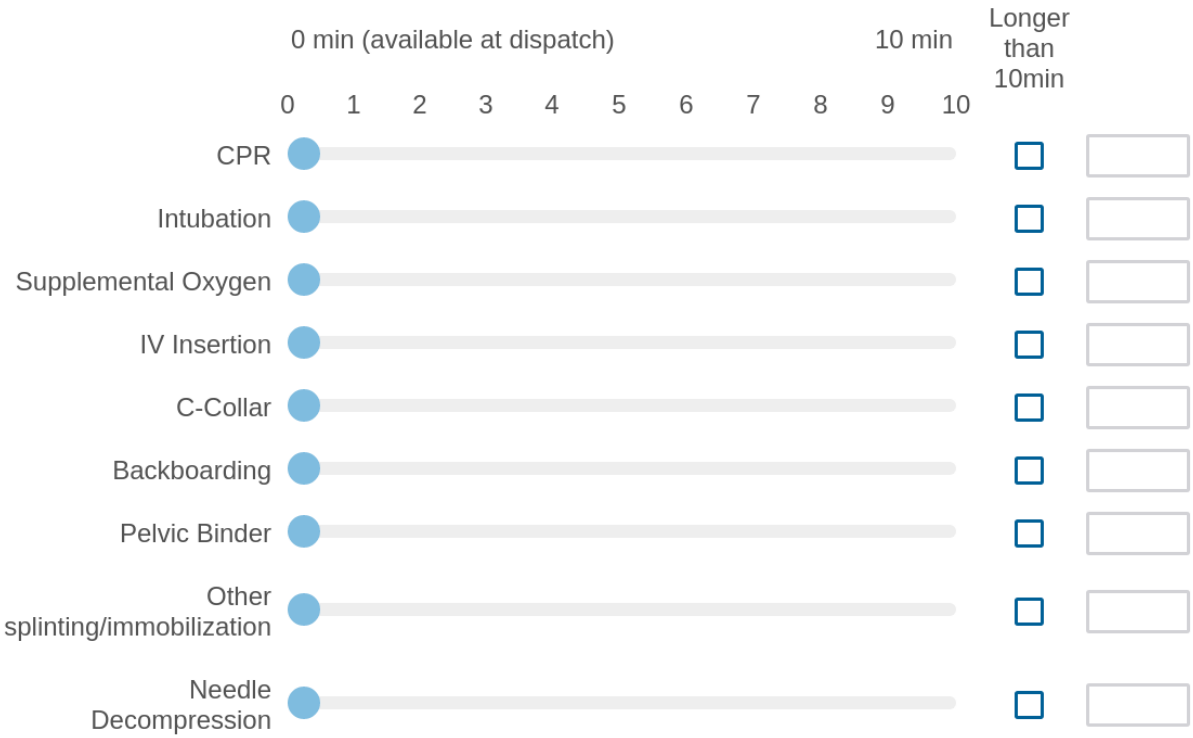


Figure 2.26: Survey form for trauma

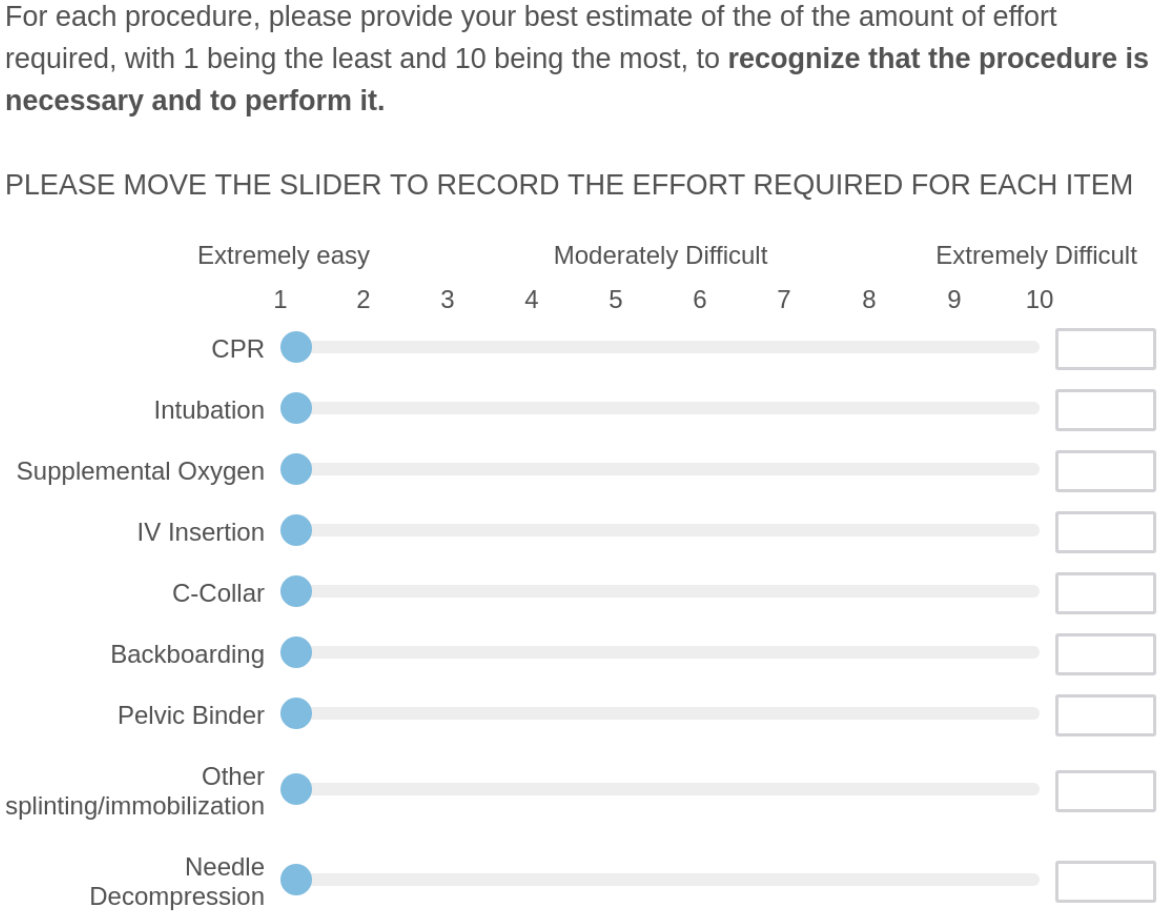


Figure 2.27: Survey form for trauma

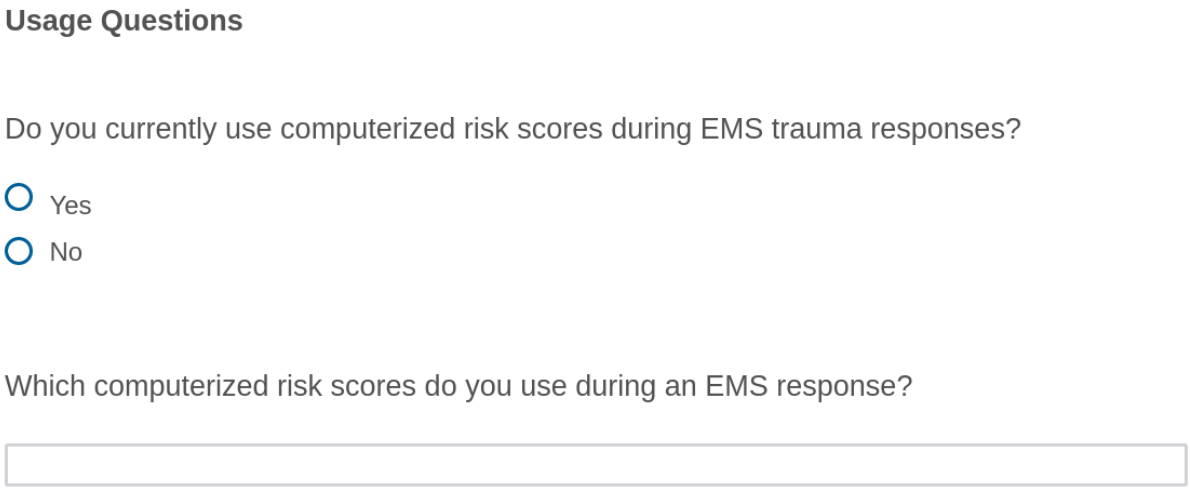


Figure 2.28: Survey form for trauma

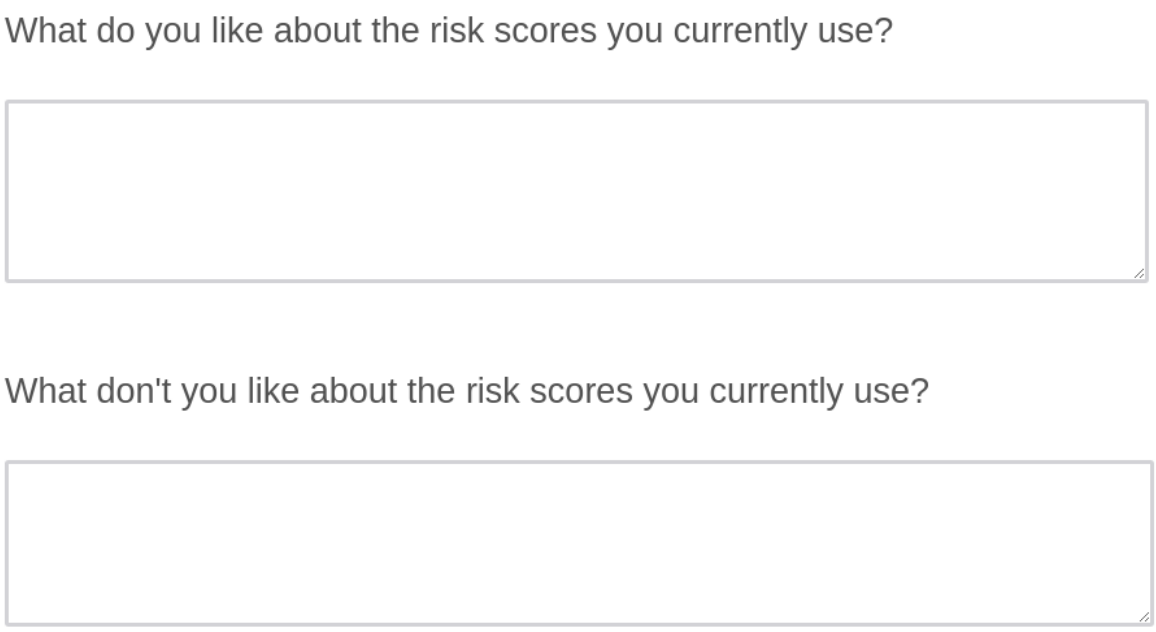


Figure 2.29: Survey form for trauma

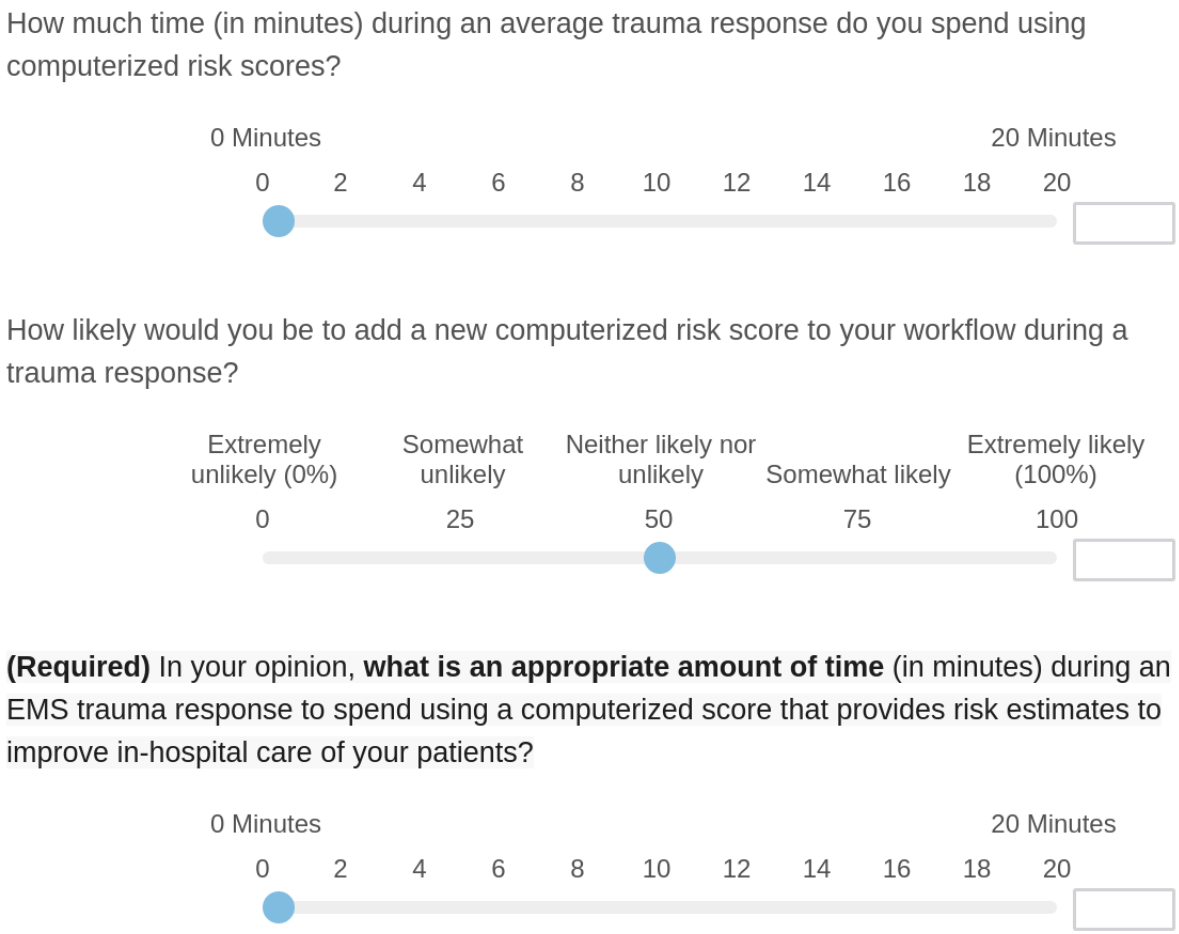


Figure 2.30: Survey form for trauma

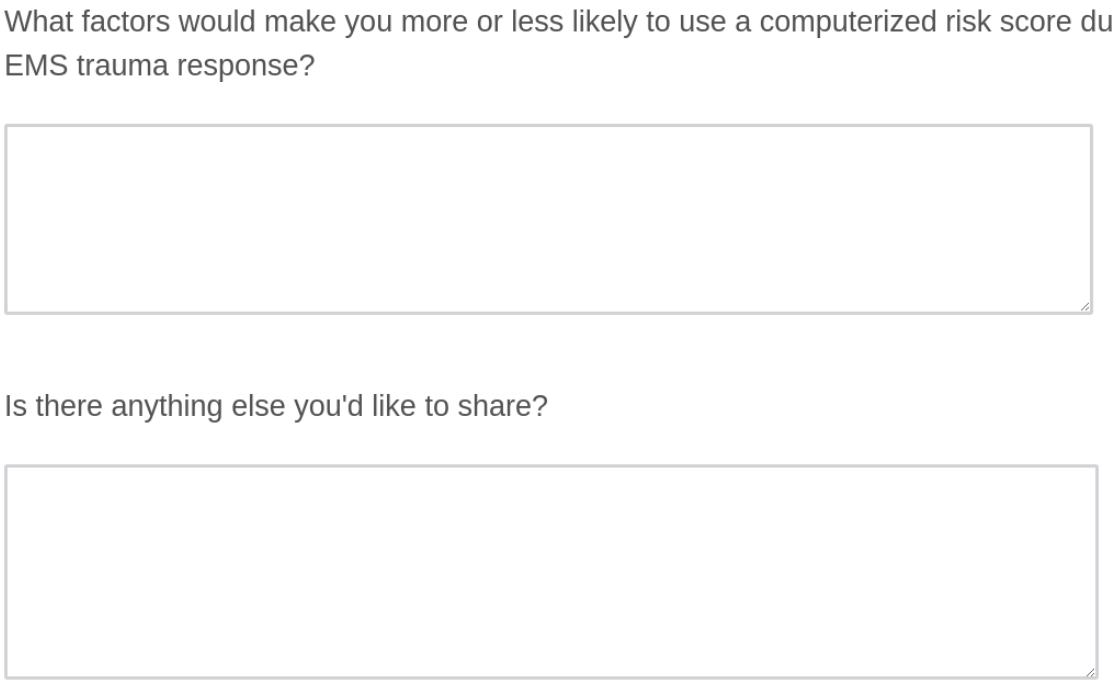


Figure 2.31: Survey form for trauma

2.5.3 EMS Provider Survey Results

#	Field		Choice Count
1	EMT-B	8.70%	2
2	AEMT	0.00%	0
3	Paramedic	69.57%	16
4	Flight paramedic	0.00%	0
5	Flight nurse	17.39%	4
6	MD	4.35%	1
			23

Figure 2.32: Survey results - certifications held by respondents

#	Field		Choice Count
1	Airlift Northwest	22.73%	5
2	Other ambulance company	4.55%	1
3	Other EMS, not an ambulance company	0.00%	0
4	Seattle Fire Medic One	72.73%	16
5	King County Medic One	0.00%	0
6	Shoreline Medic One	0.00%	0
7	Bellevue Medic One	0.00%	0
8	Redmond Medic One	0.00%	0
9	Falck	0.00%	0
10	TriMed	0.00%	0
11	AMR	0.00%	0
			22

Figure 2.33: Survey results - Agencies where respondents were employed.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	For how many years have you worked in any EMS role?	3.00	37.50	20.43	8.96	80.28	22

Figure 2.34: Survey results - years of prior EMS experience.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Age	0.00	1.00	0.25	0.34	0.12	13
2	Ethnicity	0.00	3.00	0.55	0.86	0.74	10
3	Race	0.00	1.00	0.46	0.38	0.14	10
4	Sex	0.00	1.00	0.29	0.27	0.07	13
5	Cause of injury	0.00	3.00	0.81	0.83	0.70	13
6	Protective device (seatbelts, helmets, etc)	0.10	4.00	1.45	1.37	1.88	12
7	First scene pulse	0.10	2.00	0.76	0.61	0.37	12
8	First scene respiration rate	0.10	3.80	1.03	0.97	0.94	12
9	First scene blood pressure	0.10	4.20	1.87	1.18	1.40	12
10	Lowest blood pressure while on scene	0.00	7.00	2.13	1.91	3.65	12
11	Shock index (heart rate / blood pressure)	0.20	10.00	2.38	2.50	6.26	12
12	GCS on scene	0.30	6.40	1.76	1.72	2.97	12
13	GCS - eye component	0.10	6.00	1.04	1.62	2.62	12
14	GCS - motor component	0.10	7.20	1.74	2.30	5.31	12
15	GCS - verbal component	0.10	6.10	0.93	1.60	2.56	12
16	Highest GCS while on scene	0.20	10.00	2.97	3.13	9.78	12

Figure 2.35: Survey results - Estimated time costs of gathering each feature

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Age	1.00	5.00	1.69	1.07	1.14	13
2	Ethnicity	1.00	10.00	3.92	3.17	10.07	13
3	Race	1.00	10.00	3.38	3.18	10.08	13
4	Sex	1.00	5.00	1.54	1.08	1.17	13
5	Cause of injury	1.00	4.00	2.54	0.84	0.71	13
6	Protective device (seatbelts, helmets, etc)	1.00	4.00	2.46	0.93	0.86	13
7	First scene pulse	1.00	3.00	1.77	0.58	0.33	13
8	First scene respiration rate	1.00	4.00	2.31	1.26	1.60	13
9	First scene BP	1.00	5.00	2.54	1.08	1.17	13
10	Lowest BP while on scene	1.00	10.00	3.08	2.27	5.15	13
11	Shock index (heart rate / blood pressure)	1.00	10.00	3.31	2.20	4.83	13
12	GCS on scene	1.00	10.00	3.15	2.44	5.98	13
13	GCS - eye component	1.00	3.00	1.77	0.70	0.49	13
14	GCS - motor component	1.00	4.00	2.08	0.83	0.69	13
15	GCS - verbal component	1.00	3.00	1.92	0.73	0.53	13
16	Highest GCS while on scene	1.00	10.00	2.85	2.44	5.98	13

Figure 2.36: Survey results - Estimated effort costs of gathering each feature

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	CPR	0.00	0.80	0.33	0.21	0.05	12
2	Intubation	1.00	9.90	4.46	2.86	8.17	12
3	Supplemental Oxygen	0.30	3.00	1.20	0.81	0.66	12
4	IV Insertion	0.50	5.30	2.63	1.55	2.41	12
5	C-Collar	0.50	4.00	1.81	1.03	1.05	12
6	Backboarding	0.50	6.70	3.07	2.00	4.00	12
7	Pelvic Binder	0.80	6.60	2.96	1.90	3.62	12
8	Other splinting/immobilization	0.80	7.50	3.79	1.71	2.92	12
9	Needle Decompression	0.40	8.70	3.15	2.38	5.69	12

Figure 2.37: Survey results - Time costs of performing and recording each of the following procedures.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	CPR	1.00	8.00	1.83	1.91	3.64	12
2	Intubation	1.00	8.00	4.08	2.22	4.91	12
3	Supplemental Oxygen	1.00	6.00	1.58	1.44	2.08	12
4	IV Insertion	1.00	5.00	2.42	1.19	1.41	12
5	C-Collar	1.00	3.00	1.50	0.65	0.42	12
6	Backboarding	1.00	3.00	1.83	0.90	0.81	12
7	Pelvic Binder	1.00	4.00	2.33	1.03	1.06	12
8	Other splinting/immobilization	1.00	7.00	2.58	1.55	2.41	12
9	Needle Decompression	1.00	7.00	3.50	1.55	2.42	12

Figure 2.38: Survey results - Effort costs of performing and recording each of the following procedures.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you currently use computerized risk scores during EMS trauma responses?	2.00	2.00	2.00	0.00	0.00	12

Figure 2.39: Survey results - Percentage of respondents who had used a computerized risk score (0 percent).

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	1	0.00	0.00	0.00	0.00	0.00	0

Figure 2.40: Survey results - Amount of time respondents had spent using computerized risk scores in the field (0 minutes).

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How likely would you be to add a new computerized risk score to your workflow during a trauma response?	0.00	75.00	34.08	24.61	605.41	12

Figure 2.41: Survey results - Likelihood respondents would be willing to add a new risk score to their workflow.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	1	0.00	2.00	0.83	0.80	0.64	12

Figure 2.42: Survey results - Amount of time in minutes respondents were willing to spend gathering data for a computerized risk score.

Chapter 3

Improving performance of deep learning models with axiomatic attribution priors and expected gradients

Recent research has demonstrated that feature attribution methods for deep networks can themselves be incorporated into training; these *attribution priors* optimize for a model whose attributions have certain desirable properties – most frequently, that particular features are important or unimportant. These attribution priors are often based on attribution methods that are not guaranteed to satisfy desirable interpretability axioms, such as completeness and implementation invariance. Here, we introduce attribution priors to optimize for higher-level properties of explanations, such as smoothness and sparsity, enabled by a fast new attribution method formulation called *expected gradients* that satisfies many important interpretability axioms. This improves model performance on many real-world tasks where previous attribution priors fail. Our experiments show that the gains from combining higher-level attribution priors with expected gradients attributions are consistent across image, gene expression, and health care data sets. We believe this work motivates and provides the necessary tools to support the widespread adoption of axiomatic attribution priors in many areas of applied machine learning. The implementations and our results have been made freely available to academic communities.*

3.1 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 [40, 50, 63, 64] and how they provide insight into model behavior [65, 7, 66, 67]. 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. [68], 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 [69]. 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 [50]. 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 [70]. 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 [71, 72]. 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

*This paper was joint work with co-first-authors Joseph D. Janizek and Pascal Sturmfels, as well as Scott Lundberg and Su-In Lee. It has been published as:
Gabriel G. Erion et al. "Improving performance of deep learning models with axiomatic attribution priors and expected gradients". *Nature Machine Intelligence*, published 2021, Nature Portfolio. [62].

called contextual decomposition explanation penalization (CDEP) uses a framework similar to attribution priors and penalizes explanations generated by the contextual decomposition (CD) method [73]. 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 3.2.1 and Supplementary Sections 3.5.1 and 3.5.2.

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 [74, 68, 75, 76, 77] 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 3.1a); 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 [50]. 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.

3.2 Results

3.2.1 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 Methods 3.4.1) [68, 71], 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 Methods 3.4.2 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 3.4.3, and for more details on previous attribution priors, see Methods 3.4.1. 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 [73]; 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. For further discussion of related work, including a discussion of specific cases for which our method and CDEP are best suited, see Supplement Sections 3.5.1 and 3.5.2.

3.2.2 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 [50], 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 Methods 3.4.2 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' (Figure 3.1a, 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 [50] recommends 20 to 300 gradient calls to compute attributions). This makes training with IG prohibitively slow – in fact, [71] 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 3.1 and Supplement Section 3.5.4 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 (3.1a, 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 (Figure 3.1b and Figure 3.1c). 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 (Figure 3.1a, 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 (Supplement Section 3.5.4). 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 Methods 3.4.2 and Supplement Section 3.5.3 for implementation details and pseudocode).

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 (Table 3.1, Supplement Section 3.5.4 and Supplementary Figure 3.9). The

Table 3.1: **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, see Supplement Section 3.5.4 for details on exact calculation of these metrics and exhaustive list of metrics considered).

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	—

$k = 1$ setting also appeared optimal for EG; setting $k > 1$ required more total gradient calls for convergence (Supplement Section 3.5.4 and Supplementary Figure 3.8). We also compare EG to other feature attribution methods using synthetic data benchmarks introduced in [65] (Table 3.1), 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 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. We provide more details and additional benchmarks in Supplement Section 3.5.4.

3.2.3 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 [78, 50]. 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 [79]. Although we may prefer a model with smoother attributions, existing methods only post-process attribution maps but do not change model behavior [80, 78, 81]. Such techniques may not be faithful to the original model [70]. In this section, we describe how we applied our framework to train image models with naturally smoother 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 3.4.3 for more detail). We applied this pixel smoothness attribution prior to the MNIST and CIFAR-10 datasets [74, 82]. On MNIST we trained a two-layer convolutional neural network; for CIFAR-10 we trained a VGG16 network from scratch (see Methods 3.4.4 for more details) [83]. 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 3.2 and 3.3 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

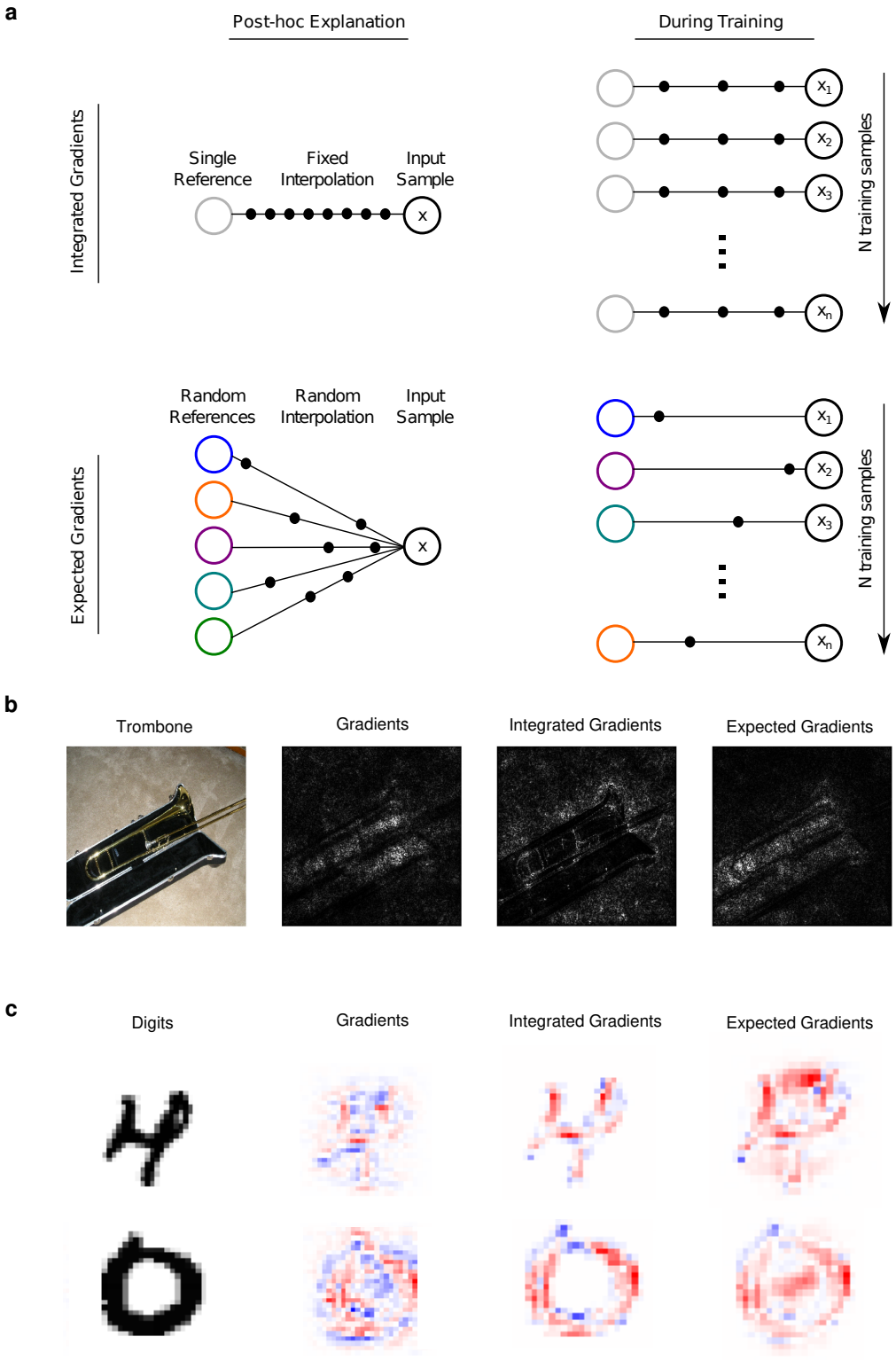


Figure 3.1: **Expected Gradients is a feature attribution method designed to be regularized during training.** (Caption continued on next page.)

Figure 3.1: (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).

underlying distribution of the training and test set can significantly reduce test accuracy [84]. 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 [68], 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 3.2 and 3.3, 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 [70, 85, 86]. 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 (Supplementary Figures 3.15-3.16). 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. For details of the EG-vs-IG comparison, results at different hyperparameter thresholds, more details on our training procedure, and additional experiments on MNIST, CIFAR-10 and ImageNet, see Supplement Sections 3.5.5, 3.5.8, 3.5.6 and 3.5.7.

3.2.4 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 [87]. 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 [88]. 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 the drug required to kill

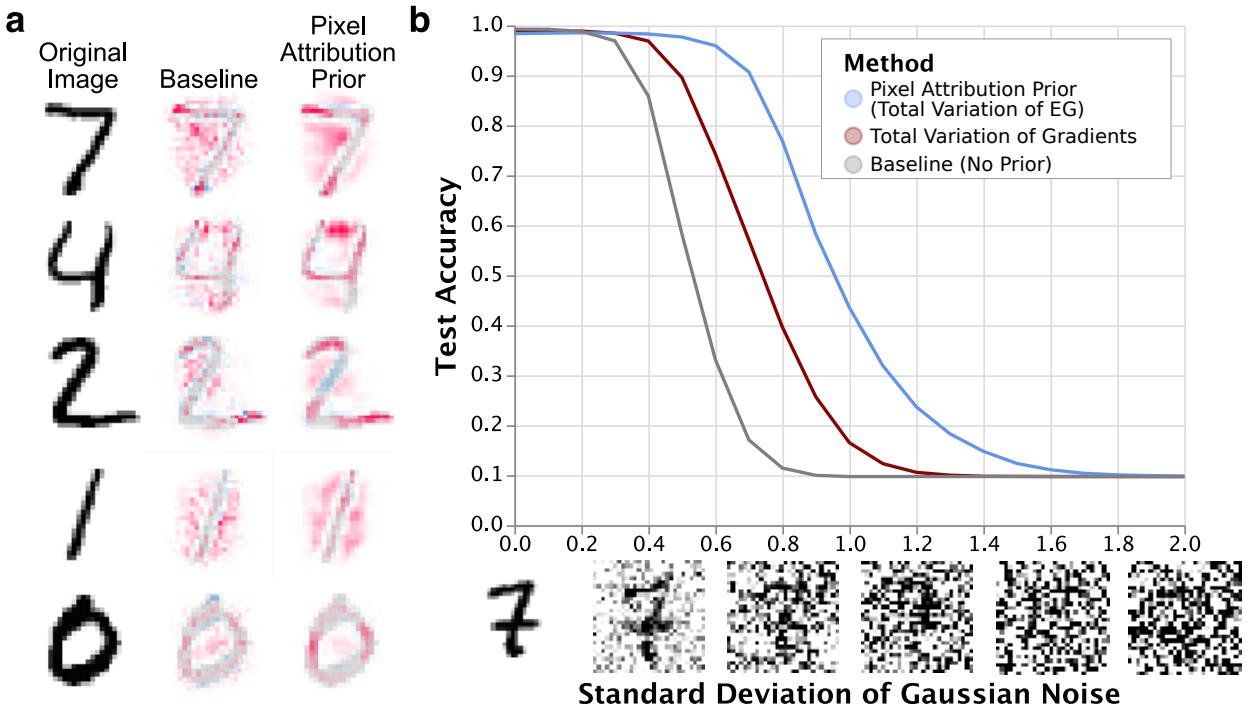


Figure 3.2: **Pixel Attribution Prior improves saliency map smoothness and increases robustness of MNIST classifier to noise. a**, EG attributions (from 100 samples) on MNIST for both an unregularized model and a model trained with an attribution prior regularized using EG. The latter achieves visually 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 [80, 81], 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.

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 [89].

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 indicates the fraction of the variance in the output explained by the model (Figure 3.4, see Methods 3.4.5 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 [68] 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 [90].

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 (Figure 3.4b). 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

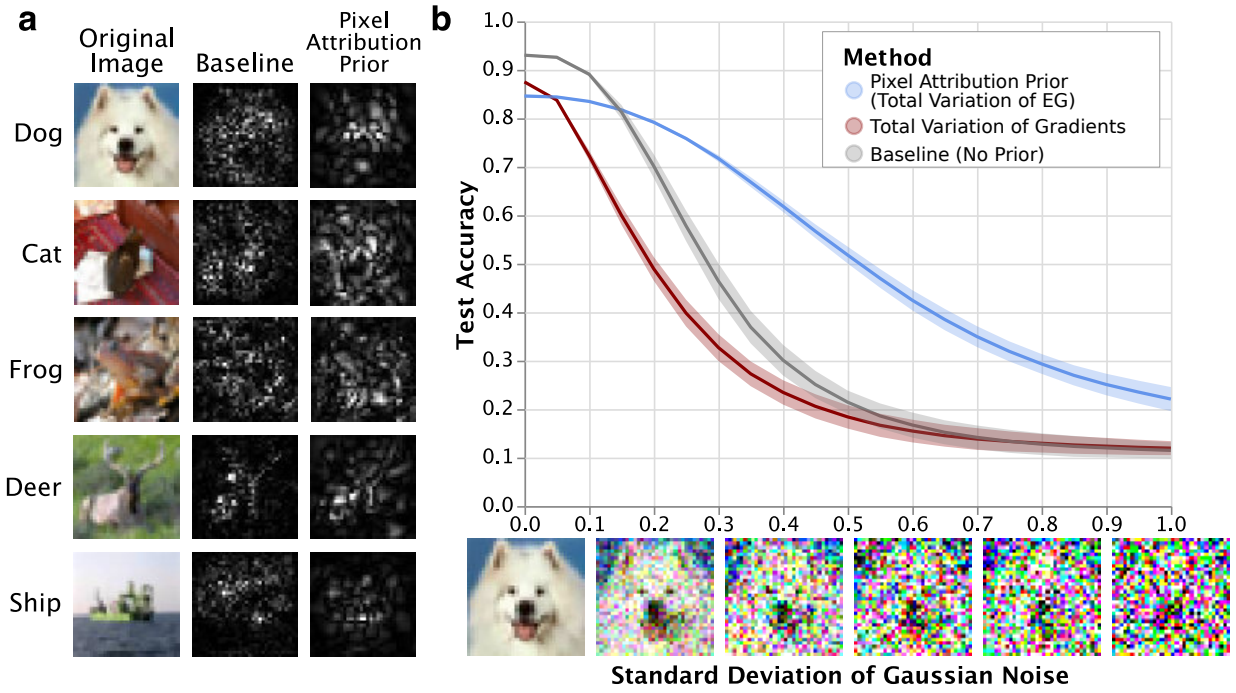


Figure 3.3: **Pixel Attribution Prior improves saliency map smoothness and increases robustness of CIFAR10 classifier to noise. a**, EG attributions (from 100 samples) on CIFAR10 for both the baseline model and the model trained with an attribution prior for five randomly selected images classified correctly 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.

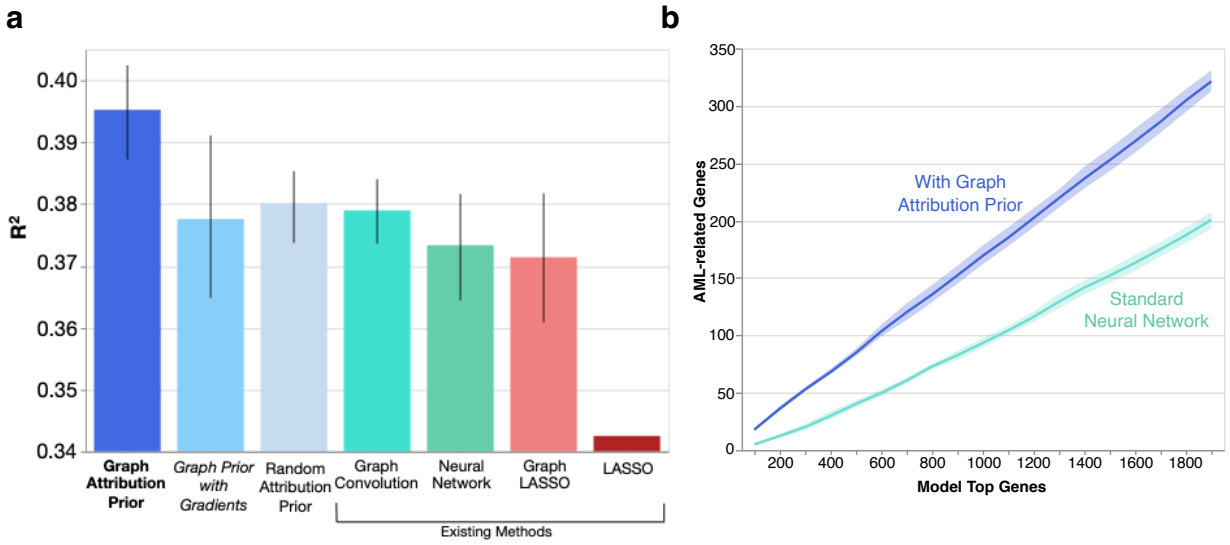


Figure 3.4: **Graph Attribution Prior improves test accuracy and biological relevance of anti-cancer drug response prediction model. a**, A neural network trained with our graph attribution prior (bold) attains the best test performance, while one trained with the same graph penalty on the gradients (italics, adapted from [68]) 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).

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 Methods 3.4.5 and Supplementary Section 3.5.9 for more detail, including the top pathways for each model) [91]. 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 [92]. 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 (see Supplementary Section 3.5.9) [93, 94]. 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.

3.2.5 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 [95, 96], while a simple existing attribution prior approach [97] 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 [98]. 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 [99, 100]. 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 [101], 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 [68, 97]. The regularization strength was tuned from 10^{-7} to 10^{5} using the validation set for all methods (see Methods 3.4.6 and Supplement Section 3.5.10).

The sparse attribution prior enables more accurate test predictions (Figure 3.5a) and sparser models (Figure 3.5c) 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 3.5d). In particular, we observe that L1 penalizing the model's gradients as in [97] rather than its EG attributions performs 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 3.5b). Details and results for other penalties, including L2, dropout, and other attribution priors, are in Supplement Section 3.5.10.

3.3 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

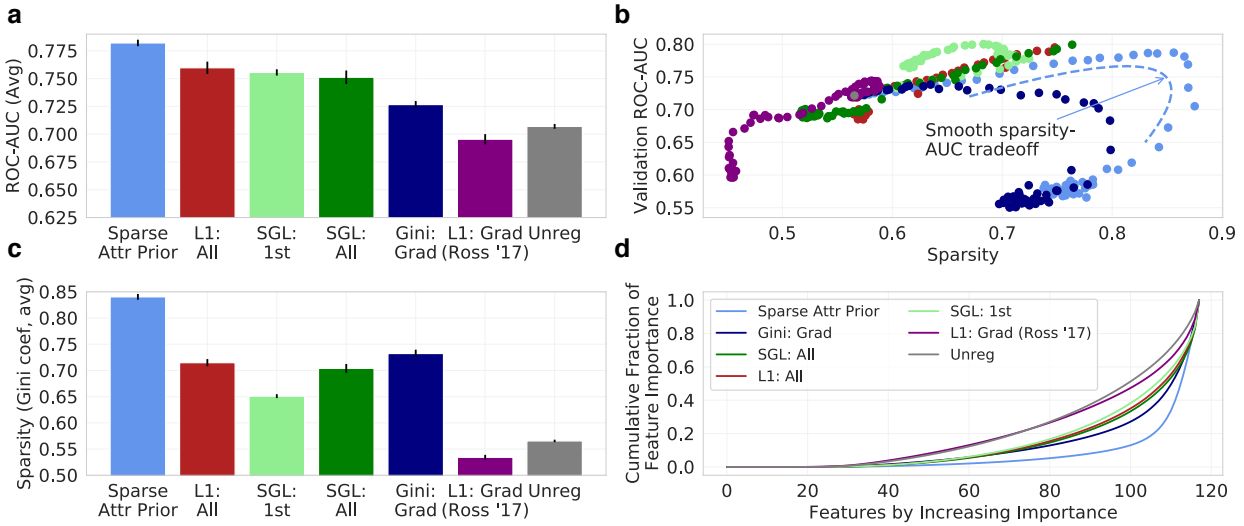


Figure 3.5: **Sparse Attribution Prior builds sparser and more accurate healthcare mortality models.** A sparse attribution prior enables more accurate test predictions (**a**) and sparser models (**c**) across 200 small subsampled datasets (100 training and 100 validation samples, all other samples 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.

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.

3.4 Methods

3.4.1 Previous attribution priors

The first instance of what we now call an attribution prior was proposed in [68], 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 [68, 71] 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 3.2 presented three newly developed attribution priors for different tasks that improve performance without requiring pre-specified attribution targets for any particular feature.

3.4.2 Expected gradients

Expected gradients is an extension of integrated gradients [50] 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 [50, 98, 102], 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 [103]. 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. For a pseudocode description of EG, see Supplement Section 3.5.3.

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 [50], 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 [50]. IG and EG both satisfy completeness due to the gradient theorem (fundamental theorem of calculus for line integrals) [50]. 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 [50].

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 [104]. 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.

3.4.3 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.

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 [105] 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 $\phi^\ell_{i,j+1} - \phi^\ell_{i,j} \sim \text{Laplace}(0, \lambda^{-1})$. [105] 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 [106, 107, 108].

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 [105].

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.

3.4.4 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 [109]. 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. See Supplement Section 3.5.6 for more details.

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 [110]. Dropout was turned off when calculating the gradients with respect to the attributions. We trained with the Adam optimizer with the default parameters ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) [111]. 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. See Supplement Section 3.5.7 for more details.

3.4.5 Biological experiments

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) [112]. 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 $= 1.7$,$p0.12$, see Supplementary Fig. 3.20).

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 [113]. 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).

Model class implementations and hyperparameters tested

LASSO. We used the scikit-learn implementation of the LASSO [114, 15]. 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 [106], with a learning rate of 10^{-5} to optimize the following loss function:

$$\mathcal{L}(w; X, y) = \|Xw - y\|_2^2 + \lambda'\|w\|_1 + \nu' w^T L_G w, \tag{3.1}$$

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 [89]. 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:

$$\mathcal{L}(\theta; X, y) = \|f_\theta(X) - y\|_2^2 + \lambda\|\theta\|_1 \tag{3.2}$$

$$\mathcal{L}(\theta; X) = \Omega_{graph}(\Phi(\theta, X)) = \nu\bar{\phi}^T L_G \bar{\phi} \tag{3.3}$$

Following [68], 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 [90]. 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.

3.4.6 Sparsity experiments

Data description and processing

Our sparsity experiments used data from the NHANES I survey [101] 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.

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. For details, see Supplement Section 3.5.10.

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 $[10^{-7}, 10^{5}]$ (Supplementary Figure 3.23). Other penalties, not displayed in the maintext experiments, are discussed in Supplement Section 3.5.10.

Maintext methods

Performance and sparsity bar plots. The performance bar graph (Figure 3.5a) 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 3.5c) 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 3.5, 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 3.5, 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. Detailed tables of the resulting p-values and test statistics T are shown in Supplement Section 3.5.10.

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. The specific version of code used in this paper is archived at [115].

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 [39] and can be accessed at `https://github.com/suinleelab/treeexplainer-study/tree/master/benchmark`

Author Contributions

G.E., J.D.J., P.S., and S.M.L. conceived the study. G.E., J.D.J.,and P.S. designed algorithms and experiments. P.S. and J.D.J. implemented core libraries for the research. G.E., J.D.J. and P.S. wrote code for and ran experiments, plotted figures, and contributed to the writing. S.M.L. contributed to the writing. S.-I.L. supervised research and method development, and contributed to the writing.

Competing Interests

The authors declare no competing interests.

Acknowledgments

The results published here are partially 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.

The authors received funding from the National Science Foundation [DBI-1759487 (S.-I.L.), DBI-1552309 (J.D.J., G.E., S.-I.L.), DGE-1256082 (S.M.L.)]; American Cancer Society [RSG-14-257-01-TBG (J.D.J.,P.S.,S.-I.L)]; and National Institutes of Health [R01AG061132 (J.D.J, P.S., S.-I.L), R35GM128638 (G.E., S.-I.L), F30HL151074-01 (G.E., S.-I.L), 5T32GM007266-46 (J.D.J, G.E.)].

3.5 Supplementary Material

3.5.1 Related Work

There have been many previous attribution methods proposed for deep learning models [40, 102, 98, 50]. We chose to extend integrated gradients because it is easy to differentiate and comes with theoretical guarantees.

Training with gradient penalties has also been discussed by existing literature. [116] introduced the idea of regularizing the magnitude of model gradients in order to improve generalization performance on digit classification. Since then, gradient regularization has been used extensively as an adversarial defense mechanism in order to minimize changes to network outputs over small perturbations of the input [76, 75, 77]. [79] make a connection between gradient-based training for adversarial purposes and network interpretability. [70] formally describe how the phenomena of adversarial examples may arise due to features that are predictive yet non-intuitive, and stress the need to incorporate human intuition into the training process.

Work on actually incorporating feature attribution methods into training is relatively recent. [117] formally describe the problem of classifiers having unexpected behavior on inputs not seen in the training distribution, like those generated by asking whether a prediction would change if a particular feature value changed. They describe an active learning algorithm that updates a model based on points generated from a counter-factual distribution. Their work differs from ours in that they use feature attributions to generate counter-factual examples, but do not directly penalize the attributions themselves. [68] introduce the idea of training models to have correct explanations, not just good performance. Their method is an instance of attribution priors in which the attribution function is gradients and the penalty function is minimizing the gradients of features known to be unimportant for each sample. [72, 71] both present attribution priors that penalize integrated gradients attributions. Our work improves on previous attribution priors in three ways. First, our broader interpretation of attribution priors allows us to instantiate three different penalty functions that encode human intuition without needing to know specific target values for each feature attribution. Second, our development of expected gradients provides a novel feature attribution method that can be regularized efficiently using a sampling procedure, allowing us to train with respect to more background references in a shorter time. Third, we empirically show that using an axiomatic method like expected gradients yields substantially better results than simple gradient-based priors. Overall, flexible attribution priors that use axiomatic feature attribution methods lead to more interpretable models with better performance than previous approaches.

3.5.2 Comparison with Contextual decomposition explanation penalization

In addition to the prior methods we compare our method with in the main text, we also compare to a contemporary method called contextual decomposition explanation penalization (CDEP) [73]. This framework is closely related to attribution priors, with small changes:

$$\theta = \text{argmin}_\theta \mathcal{L}(\theta; X, y) + \lambda \sum_i \Omega_c(\Phi_c(\theta, x_i, s_i))$$

where Φ_c now represents the attribution method contextual decomposition [118] rather than our attribution method, expected gradients. Importantly, Φ_c produces an attribution for a single *group* of features, rather than an attribution for each individual feature. We call the feature set whose attributions are desired for a given sample s_i. Ω_c takes the more limited form of $||\beta_i - \Phi_c(\theta, x_i, s_i)||_1$ where β_i represents a target attribution for the feature set s_i.

Our approach offers substantial benefits when compared to CDEP. First, our method is significantly easier to use in practice. The attribution method Contextual Decomposition can not simply be applied to any architecture of network and requires custom decomposition rules for each type of layer.[†] This means it may be difficult or impossible to use some architectures (i.e., attention-based networks) with Contextual Decomposition. Training attribution priors using expected gradients, on the other hand, does not require any knowledge of or adaptation to the specific network architecture in question. Any network with gradients that can be calculated by TensorFlow or PyTorch will automatically work with our approach.

[†]As noted by the authors on the project's Github repository, "the current CD implementation doesn't always work for all types of networks... you may need to write a custom function that iterates through the layers of your network." [119]

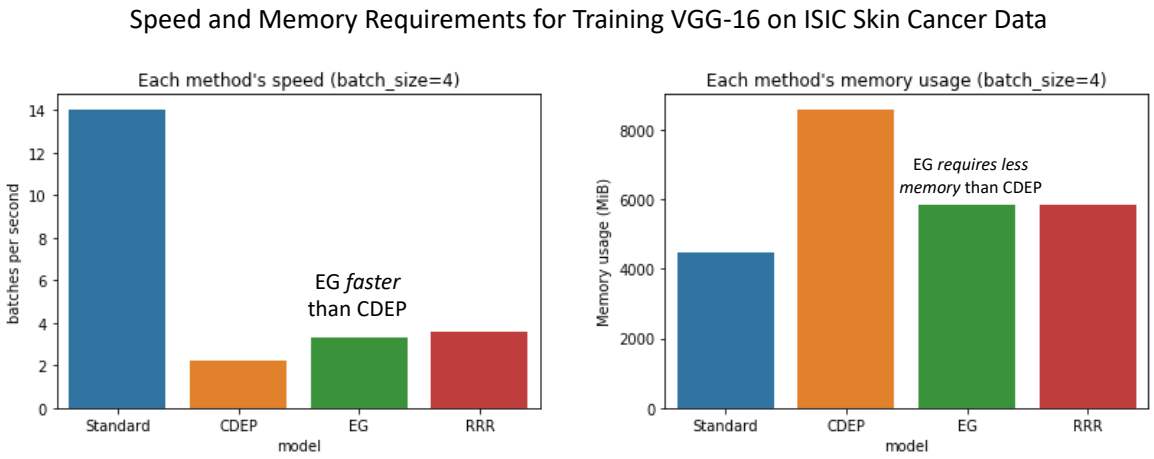


Figure 3.6: When training the full VGG-16 model, EG outperforms CDEP in terms of batches per second and memory requirement on ISIC Skin Cancer dataset.

Furthermore, we find that EG is more computationally efficient than CDEP and that the CDEP paper substantially underestimates the computational efficiency of our method. The paper compares CDEP, EG, and the Right for the Right Reasons (RRR) method [68] on three datasets – Grayscale Decoy, Color MNIST, and ISIC Skin Cancer. Results on the first two datasets are inconclusive: CDEP outperforms both EG and RRR on Color MNIST, while EG and RRR outperform it on Grayscale Decoy. For the third dataset, however, the authors only compare to RRR, claiming that EG is too slow and memory-intensive to run on the dataset.[‡]

In fact, we show that EG runs faster and with less memory usage than CDEP on the ISIC dataset. Using the code available in the CDEP authors' publicly-available repository [119], we downloaded the data to reproduce their skin cancer data experiment. We evaluated the amount of memory and time required to train each batch of image data on the full VGG-16 network with each method. We were limited to batch sizes of 4 images for all methods because this was the largest batch size that could be run with CDEP on an NVIDIA GEFORCE RTX 2080 Ti GPU (EG and RRR both could accommodate larger batch sizes).

We found that while all three methods are slower than training without any kind of attribution prior, EG and RRR both train faster than CDEP (see Supplemental Figure 3.6). Furthermore, EG and RRR both require less GPU memory than CDEP. Finally, while the CDEP paper claims that EG requires substantially more memory and processing time than RRR, our results show that EG and RRR are in fact nearly identical with regard to speed and memory.

The authors of the CDEP paper are able to report substantially faster performance than we observe here because, in their experiments, they freeze the feature extracting layers of the VGG-16 (all layers except the final fully connected layers), and pre-compute CD attributions and image features for each image up to the end of the frozen layers in the network. When CDEP is run in identical conditions to EG and RRR, without these preprocessing steps, its performance is substantially worse. While the preprocessing steps used in CDEP may be useful ways to improve training speed, it is not accurate to conclude that contextual decomposition is a faster explanation method or that incorporating it into training as in CDEP is inherently more computationally efficient than EG – in fact, we show the opposite is the case, and that the claimed speed benefits of CDEP are due to preprocessing steps which may not be applicable in all cases.

It is also possible that the CDEP paper overestimates the resource requirements of EG and attribution priors by choosing the wrong hyperparameter k (number of references). As we show in Sections 2.2 and 2.3, $k = 1$ is sufficient to train high quality models in our image experiments. A setting much higher than $k = 1$ could dramatically slow down EG without improving performance.

Finally, we observe that CDEP would be completely unsuitable as an attribution method for the types of

[‡]"We were not able to compare against the method recently proposed in Erion et al., 2019 due to its prohibitively slow training and large memory requirements."[73]

priors we proposed in our work. Each prior suggested in our work requires attributions for each individual feature, and using EG we can attain those attributions during training at the cost of a single backward pass through the original model and first-order gradients with respect to each input. To attain individual feature-level attributions using Contextual Decomposition would require an additional d forward passes for each batch during training (where d is the number of input features), as well as a backward pass through each of the d resulting attributions, and thus would be entirely computationally infeasible.

The CDEP framework is very efficient when attributions are desired for a *small number of groups of features*. While it is completely impractical to calculate during training if *many individual feature attributions* are desired, CDEP will likely outperform EG in terms of efficiency in the case when several conditions are met: 1) attributions are only necessary for a small number of groups of features, 2) only part of the network needs to be tuned, and 3) the network is of an architecture that CD can be applied to. Furthermore, when the priors that it is desirable to enforce are easily expressed as some function of groups of features, experimental results in the CDEP paper indicate that their method may outperform attribution methods that generate explanations for single features.

In conclusion, attribution priors offer several potential advantages over CDEP, including compatibility with both a greater variety of models and a greater variety of priors. In addition, the strongest criticisms leveled at attribution priors by the CDEP paper seem to stem from implementation differences rather than real deficiencies in our method.

3.5.3 Algorithm for training attribution priors with Expected gradients

We assume access to the following subroutines:

- RAND(a, b, n), which returns a vector of n random floats between a and b.
- RANDOMSAMPLE(X, n), which returns a matrix consisting of n random rows of X, sampled with replacement.
- GRAD(o, i), which returns the gradients of outputs o with respect to inputs i.
- MEAN(X, d), which returns the mean of multi-dimensional array X along dimension d.
- GETBATCH(X, y, b), which returns a batch of b samples from data X and labels y.
- LOSS(f, X, y), which returns the desired prediction loss given a model f, data X, and labels y
- APLOSS(Φ), which returns the value of the attribution prior loss given attributions Φ
- UPDATE(θ, G), which takes as input current parameters θ and gradients G w.r.t. those parameters, and returns new parameters after a step with some optimizer (gradient descent, RMSProp, Adagrad, Adam, etc)

We also use Python-style array indexing; that is, $X[:, i]$ means slice i of multi-dimensional array X along axis 1.

Our first algorithm, EG, takes a batch of samples, a reference set, a model, and a number of interpolation steps k and returns a matrix of attributions of the same shape as the sample batch.

Algorithm 1 EG

procedure EG(B,R,f,k) ▷ B sample batch, R reference set, f model, k # interpolation steps
  n, p =SHAPE(B)
  v = zeros of dimension $n \times k$ ▷ Stores model output for each interpolation
  g = zeros of dimension $n \times p \times k$ ▷ Stores gradients for each interpolation
  for $i \leftarrow 0$ to k **do**
    α = RAND$(0, 1, n)$ ▷ Random interpolation amounts
    S = RANDOMSAMPLE(R,n) ▷ Draw reference batch from R
    $I = (1 - \alpha) \odot S + \alpha \odot B$ ▷ Interpolation between S and B
    $v[:, i] = f(I)$ ▷ Evaluate model at interpolated points I
    $g[:, :, i] = (B - I) \times$ GRAD(v, I) ▷ Scaled gradients of model output w.r.t. I
  return MEAN$(g, 2)$ ▷ Expectation over scaled gradients

Our second algorithm, APRIOR-EG, trains a model using the attribution prior APLOSS. It follows the same general pattern as gradient descent (and its variants) on the loss, but also calculates EG attributions for each batch and adds a loss function on the EG attributions to the total loss at each step.

Algorithm 2 APRIOR-EG

procedure APRIOR-EG(X,y,f,θ,s,b,k,λ)
  ▷ X data, y labels, f model, θ params, s # epochs, b batch size, k # interp steps, λ prior strength
  for $i \leftarrow 0$ to s **do**
    for $j \leftarrow 0$ to b **do**
      X_b, y_b =GETBATCH(X, y, b)
      L = LOSS(f, X_b, y_b) ▷ Loss on training labels
      Φ = EG (X_b, X, f, k) ▷ Get attributions
      Ω = APLOSS(Φ) ▷ Attribution prior loss
      $T = L + \lambda \times \Omega$ ▷ Total loss
      G = GRAD(T, θ) ▷ Total loss gradients w.r.t. model parameters
      θ = UPDATE(θ, G) ▷ Take step with RMSProp, Adam, etc.
  return θ

3.5.4 Benchmarking Expected Gradients

Sampling convergence

Since expected gradients reformulates feature attribution as an *expected value* over two distributions (where background samples x' are drawn from the data distribution and the linear interpolation parameter α is drawn from $U(0, 1)$), we wanted to ensure that we are drawing an adequate number of background samples for convergence of our attributions when benchmarking the performance of our attribution method. Since our benchmarking code was run on the Correlated Groups 60 synthetic dataset, as a baseline we explain all 1000 samples of this data sampling the full dataset (1000 samples) as background samples. To assess convergence to the attributions attained at this number of samples, we measure the mean absolute difference between the attribution matrices resulting from different numbers of background samples (see Figure 3.7). We empirically find that our attributions are well-converged by the time 100-200 background samples are drawn. Therefore, for the rest of our benchmarking experiments, we used 200 as the number of background samples. During training, even using the lowest possible setting of $k = 1$, we end up drawing far more than 200 background samples over the course of an epoch (order of magnitude in the tens of thousands, rather than hundreds).

We additionally considered a simple experiment using synthetic data to assess the impact of the number of background samples and interpolation points on the rate of convergence of the training process of a model (rather than the convergence of a single explanation). We define a synthetic dataset with two binary features, x_1 and x_2. In the training data, x_1 and x_2 are perfectly correlated. In the test data, they are uncorrelated. We want to model a third variable that is defined $y = x_1$. Given the high degree of correlation between the two features in the training data, we can then take a neural network which depends equally on both features.

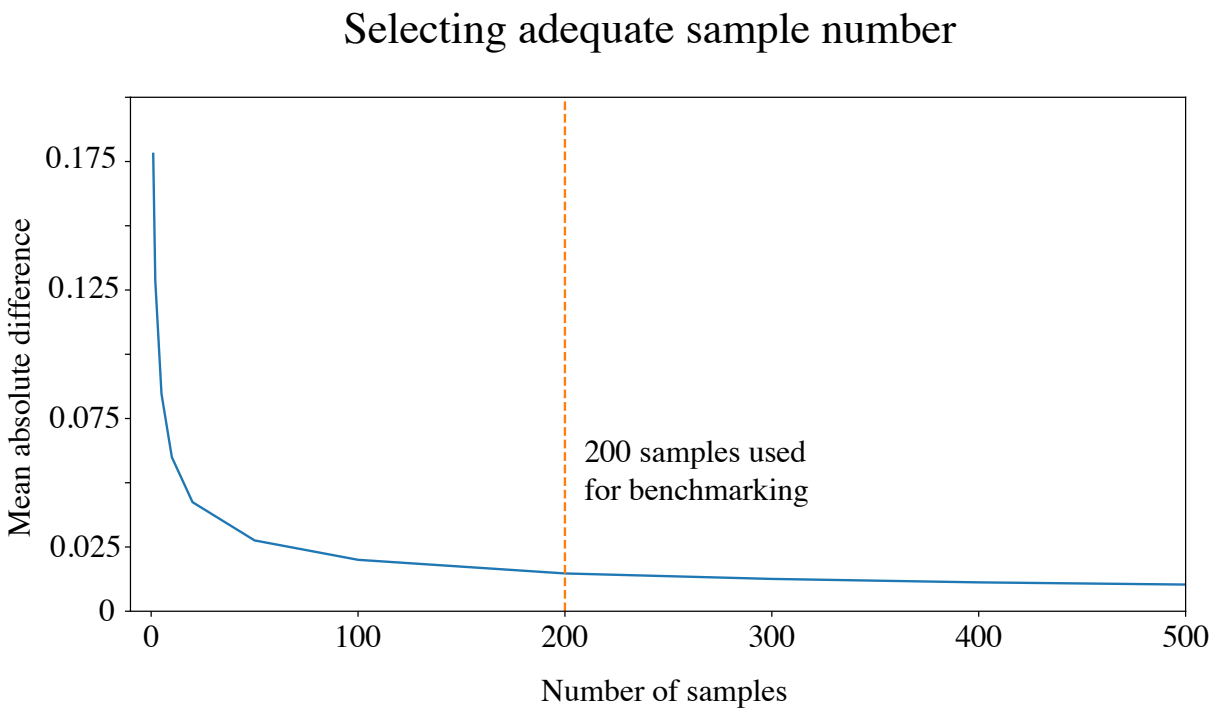


Figure 3.7: Feature attribution values attained using expected gradients converge as the number of background samples drawn is increased.

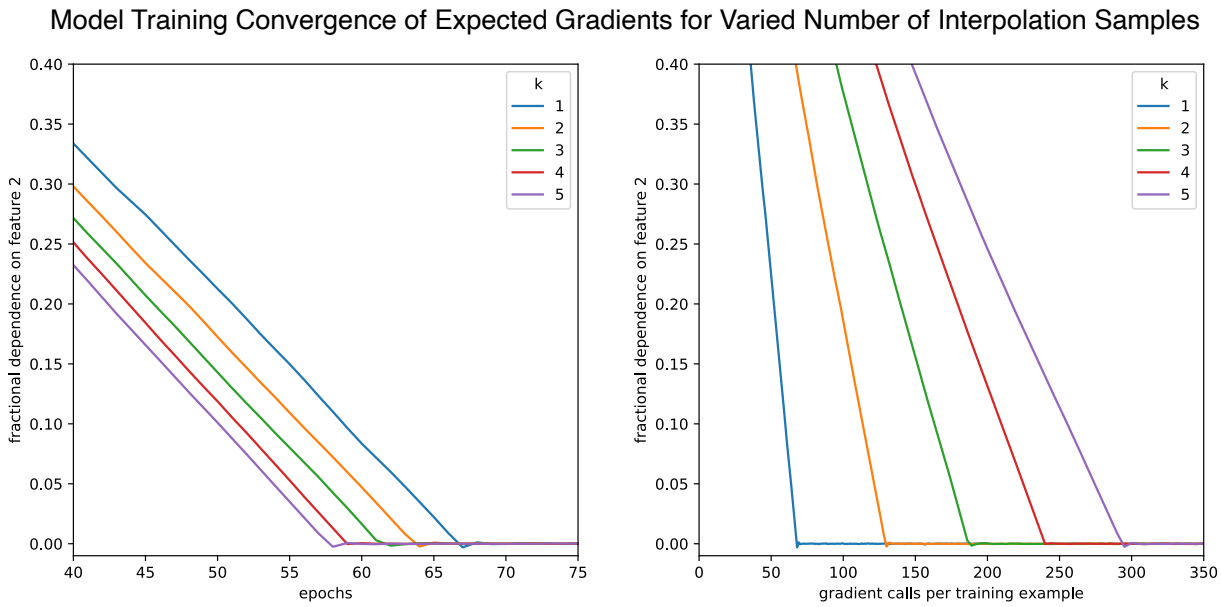


Figure 3.8: Convergence of model training process as a function of number of interpolation samples. (Left) Convergence plotted by number of epochs. (Right) Convergence plotted by number of additional gradient calls per training sample.

The network we use has a single layer with sigmoid activation. Our goal of training with an attribution prior is to remove the model's dependence on x_2, so we penalize the average magnitude of the expected gradients attribution for x_2 during training while simultaneously optimizing a binary cross entropy loss on the training labels. We measure how many training epochs are necessary to remove the model's ground truth dependence on x_2, as measured by the change in model output as the feature is toggled on and off, averaged over 100 evenly spaced values of x_1 between 0 and 1. For Expected Gradients in this experiment, we use a single reference background because the all-0s vector represents a natural background for this task.

We see that as the number of interpolation points is increased from one to five, the number of epochs required to remove the dependence of x_2 decreases. This makes sense, as increasing the number of interpolation points will give a better estimate of the attribution to be penalized during training, and consequently require less training epochs to successfully regularize this attribution. However, we also observe that the decrease in required number of epochs is less than linear with respect to the number of interpolation samples. When we plot the same data as a function of the number of additional gradient calls required per training example, which better reflects the actual amount of training time required, we can see that $k = 1$ is the most efficient setting.

Finally, we compared the convergence of our approach (expected gradients), to integrated gradients [71] and gradients [68] using the same experimental setup described above. For the comparison of expected gradients to integrated gradients, we wanted to isolate the benefit of random sampling of alpha as opposed to sampling a small number of fixed points along the interpolation path as proposed by [71]. We therefore used a fixed background of the all-zeros vector for both integrated gradients and expected gradients. For gradients, we examined penalizing both the gradients taken with respect to the model's raw output as well as the gradients taken with respect to the normalized log probability (we note that the latter is the approach actually employed in [97]). We see that expected gradients converges significantly more quickly than integrated gradients even when integrated gradients uses more additional gradient calls per training sample, owing to the fact that over the course of training for multiple epochs expected gradients can sample more unique points along the integration path than integrated gradients (which will always pick the same set of points). We also see that expected gradients converges more quickly than gradients, and that the log probability gradient is more effective than the raw output gradient, and may be less vulnerable to saturation.

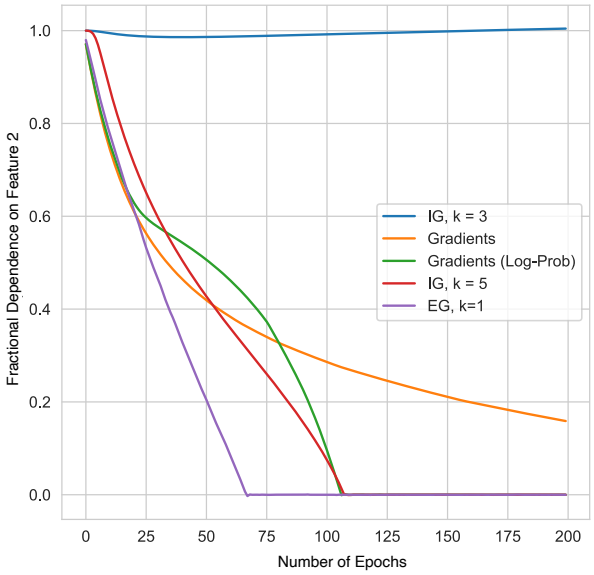


Figure 3.9: Convergence of model training process compared between expected gradients and other methods. Comparisons are with integrated gradients at several numbers of interpolation points, and gradients (where we consider both the gradients taken with respect to the model's raw output as well as the gradients taken with respect to the normalized log probability). Note that in this experiment we use a single background reference for EG, so the performance benefit is solely attributable to sampling more unique points along the interpolation path during training.

To summarize the convergence rates of the different methods, we measure the area under the curve for each of the methods in Figure 3.9 and divide by the maximal possible area (fractional dependence 1.0 for all 200 epochs) so that the areas range between 0 (perfect) and 1 (worst).

Benchmark evaluation metrics

To compare the performance of expected gradients with other feature attribution methods, we used the benchmark metrics proposed in [65] and implemented in the SHAP package [§]. These metrics were selected as they capture a variety of recent approaches to quantitatively evaluating feature importance estimates. For example, the Keep Positive Mask metric (KPM) is used to test how well an attribution method can find the features that lead to the greatest increase in the model's output. This metric progressively removes features by masking with their mean value, in order from least positive impact on model output to most positive impact on model output, as ranked by the attribution method being evaluated. As more features are masked, the model's output is increased, creating a curve. The KPM metric measures the area under this curve (larger area corresponds to better attribution method). In addition to the KPM metric, 17 other similar metrics (e.g. Remove Absolute Resample, Keep Negative Impute, etc.) were used (see supplementary material of [65] for more details on benchmark metrics). For all of these metrics, a larger number corresponds to a better attribution method. In addition to finding that Expected Gradients outperforms all other attribution methods on nearly all metrics tested for the dataset shown in Table 1 in the main text (the synthetic Correlated Groups 60 dataset proposed in [65]), we also tested all 18 metrics on another dataset proposed in the same paper (Independent Linear 60) and find that Expected Gradients is chosen as the best method by all metrics in that case as well (see Table 3.3). The Independent Linear 60 dataset is comprised of 60 features, where each feature is a 0-mean, unit variance gaussian random variable plus gaussian noise, and the label to predict is a linear function of these features. The Correlated Groups 60 dataset is essentially the same, but now certain groups of 3 features have 0.99 correlation.

[§]https://github.com/slundberg/shap

Table 3.2: Results from benchmark software on synthetic data with correlated features. Larger numbers mean a better feature attribution method for all metrics. Metric abbreviations are: K (Keep), R (Remove); P (Positive), N (Negative), A (Absolute); M (Mean masking), R (Resample masking), I (Impute masking). For example, KPM corresponds to the "Keep Positive with Mean masking" metric. Each model is trained on 900 samples and tested using 100 samples. Expected gradients is the best attribution method in all but one example ($p = 7.2 \times 10^{-5}$, one-tailed Binomial test). See Supplement Section 3.5.4 for details.

Method	KPM	KPR	KPI	KNM	KNR	KNI	KAM	KAR	KAI
Expected Grad.	**3.731**	**3.800**	**3.973**	**3.615**	**3.551**	**3.873**	**0.906**	**0.903**	0.919
Integrated Grad.	3.667	3.736	3.920	3.543	3.476	3.808	0.905	0.899	**0.920**
Gradients	0.096	0.122	0.099	0.076	-0.112	0.052	0.838	0.823	0.887
Random	0.033	0.106	0.077	-0.012	-0.093	-0.053	0.593	0.583	0.715
Method	RPM	RPR	RPI	RNM	RNR	RNI	RAM	RAR	RAI
Expected Grad.	**3.612**	**3.575**	**3.525**	**3.759**	**3.830**	**3.683**	**0.897**	**0.885**	**0.880**
Integrated Grad.	3.539	3.503	3.365	3.687	3.754	3.543	0.872	0.859	0.822
Gradients	0.035	-0.098	-0.020	0.110	0.105	0.108	0.729	0.712	0.616
Random	-0.053	-0.100	-0.106	0.034	0.092	0.111	0.400	0.400	0.275

Table 3.3: Benchmark on Independent Linear 60 dataset

Attribution Method	KPM	KPR	KPI	KNM	KNR	KNI	KAM	KAR	KAI
Expected Gradients	**4.096**	**4.179**	**4.264**	**4.014**	**3.835**	**4.153**	**0.941**	**0.946**	**0.938**
Integrated Gradients	4.055	4.112	4.176	3.949	3.753	4.070	**0.941**	0.945	**0.938**
Gradients	0.044	0.107	0.029	0.155	-0.150	0.172	0.902	0.905	0.902
Random	-0.152	0.102	-0.152	0.111	-0.126	0.060	0.470	0.482	0.438
Attribution Method	RPM	RPR	RPI	RNM	RNR	RNI	RAM	RAR	RAI
Expected Gradients	**4.079**	**3.941**	**4.210**	**4.203**	**4.260**	**4.356**	**0.992**	**0.977**	**1.019**
Integrated Gradients	4.013	3.854	4.113	4.157	4.186	4.259	0.973	0.966	0.995
Gradients	0.110	-0.125	0.133	0.057	0.080	0.041	0.947	0.936	0.985
Random	0.012	-0.124	0.059	0.035	0.101	0.070	0.504	0.521	0.527

For attribution methods to compare, we considered expected gradients (as described in the main text), integrated gradients (as described in [50]), gradients, and random.

3.5.5 Expected Gradients on ImageNet

One unfortunate consequence of choosing an arbitrary baseline point for methods like integrated gradients is that the baseline point by definition is unimportant. That is, if a user chooses the constant black image as the baseline input, then purely black pixels will not be highlighted as important by integrated gradients. This is true for any constant baseline input. Since expected gradients integrates over a dataset as its baseline input, it avoids forcing a particular pixel value to be unimportant. To demonstrate this, we use the inception v4 network trained on the ImageNet 2012 challenge [120, 121]. We restore pre-trained weights from the Tensorflow Slim library [122]. In Figure 3.10, we plot attribution maps of both expected gradients and integrated gradients as well as raw gradients. Here, we use the constant black image as a baseline input for integrated gradients. For both attribution methods, we use 200 sample/interpolation points. The figure demonstrates that integrated gradients fails to highlight black pixels.

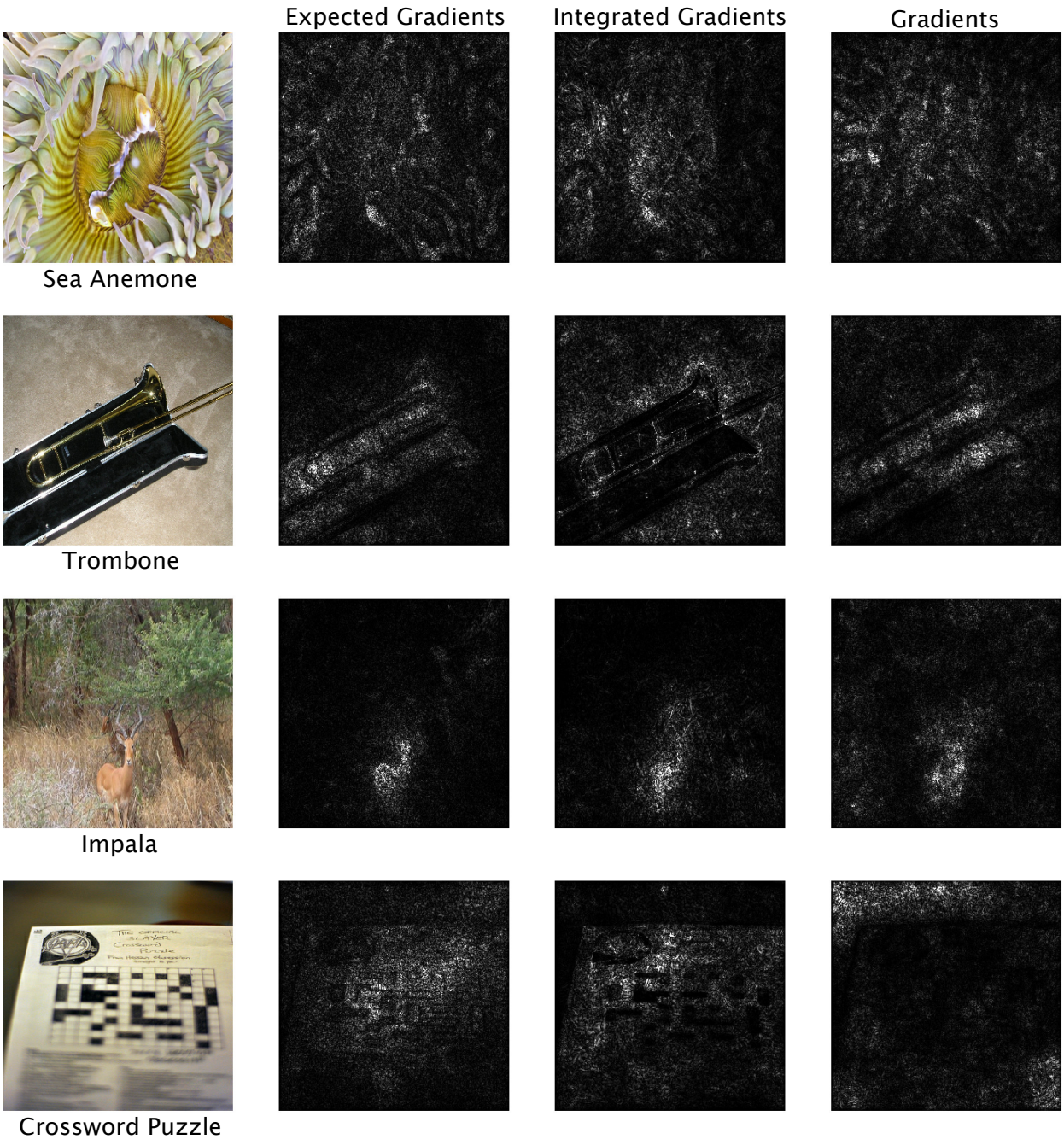


Figure 3.10: A comparison of attribution methods on ImageNet. Integrated gradients fails to highlight black pixels as important when black is used as a baseline input.

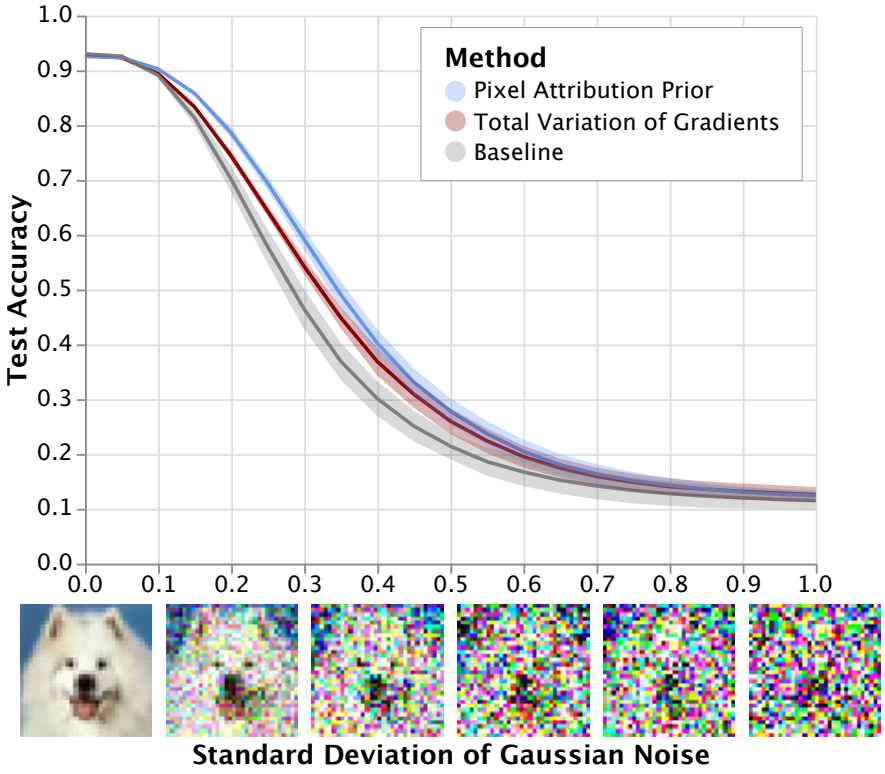


Figure 3.11: Robustness to noise on CIFAR-10 with a stricter λ threshold. Here, there is little difference in test accuracy on the original test set between the baseline and the image attribution prior model (0.930 ± 0.002 for the baseline vs. 0.925 ± 0.002 for the pixel attribution prior). Both the image attribution prior model and the gradient-based model afford small improvements in robustness compared to the baseline. As in the main text, results here are the mean and standard deviation across 5 random initializations.

3.5.6 CIFAR-10 Experiments

Choosing Lambda

In the main text, we demonstrated the robustness of the image attribution prior model with λ chosen as the value that minimized the total variation of attributions while keeping test accuracy within 10% of the baseline model. This corresponds to $\lambda = 0.001$ for both gradients and expected gradients if we search through 20 values logarithmically spaced in the range $[10^{-20}, 10^{-1}]$. If instead, we choose the λ that minimizes total variation of attributions while keeping test accuracy equivalent to the baseline model (within 1%), we see that both the attribution prior and regularizing the gradients provides modest robustness to noise. This corresponds to $\lambda = 0.0001$ for both gradients and expected gradients. We show this result in Figure 3.11.

For both the gradient-based model and the image attribution prior model, we also plot test accuracy and total variation of the attributions (gradients or expected gradients, respectively) in Figure 3.12. The λ values we use correspond to the immediate two values before test accuracy on the original test set breaks down entirely for both the gradient and image attribution prior model.

3.5.7 MNIST Experiments

Results

We choose λ by sweeping over values in the range $[10^{-20}, 10^{-1}]$. We choose the λ that minimizes the total variation of attributions such that the test error is within 1% of the test error of the baseline model, which corresponds to $\lambda = 0.01$ for both the gradient model and the pixel attribution prior model. In Figure 3.13, we plot the robustness of the baseline, the model trained with an attribution prior, and the model trained by

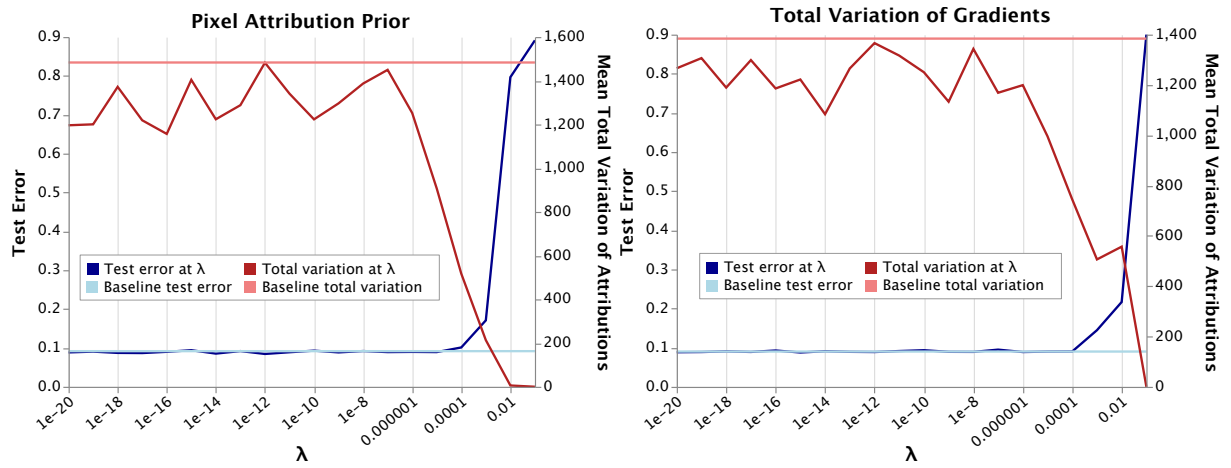


Figure 3.12: Plotting the trade-off between accuracy and minmizing total variation of expected gradients (left) or gradients (right). For both methods, there is a clear elbow point after which test accuracy degrades to no better than random. The total variation of attributions is judged based on the attribution being penalized: expected gradients for the left plot, gradients for the right plot.

penalizing the total variation of gradients. We find that on MNIST, penalizing the gradients does similarly to penalizing expected gradients. We also find that it is easier to achieve high test set accuracy and robustness simultaneously.

Attribution Maps

In Figure 3.13 we plot the attribution maps of the baseline model compared to the model regularized with an image attribution prior. We find that the model trained with an image attribution prior more smoothly highlights the digit in the image.

Other attribution priors

We also re-ran the MNIST attribution prior experiments with several other attribution methods. We used a base convolutional network with 2 convolution layers (first layer consisting of 32 size-3 stride-1 convolutions, second layer consisting of 64 size-3 stride-1 convolutions) and two fully-connected layers (size 9216->128 and 128->10) with dropout before each fully-connected layer ($p = 0.25$ and $p = 0.5$ respectively) [¶]. We then trained models with the following regularizers and attribution priors:

- **Unregularized:** This model was a simple convolutional network with no attribution prior.
- **Gradients (Logits):** This model penalized gradients of the logits of the correct label's output.
- **Gradients (Log-Probs):** This model penalized gradients of the output log-probability corresponding to the correct label.
- **IG:** This model penalized integrated gradients attributions with respect to the correct label with the all-zeros reference and $k = 10$.
- **Single-Reference EG:** This model penalized single-reference EG attributions as described in maintext Section 3.2.2 with respect to the correct label with the all-zeros reference and $k = 10$.
- **Multi-Reference EG:** This model penalized expected gradients attributions with respect to the correct label with a uniform reference distribution over the training set and $k = 10$.

[¶] `https://github.com/pytorch/examples/tree/master/mnist`

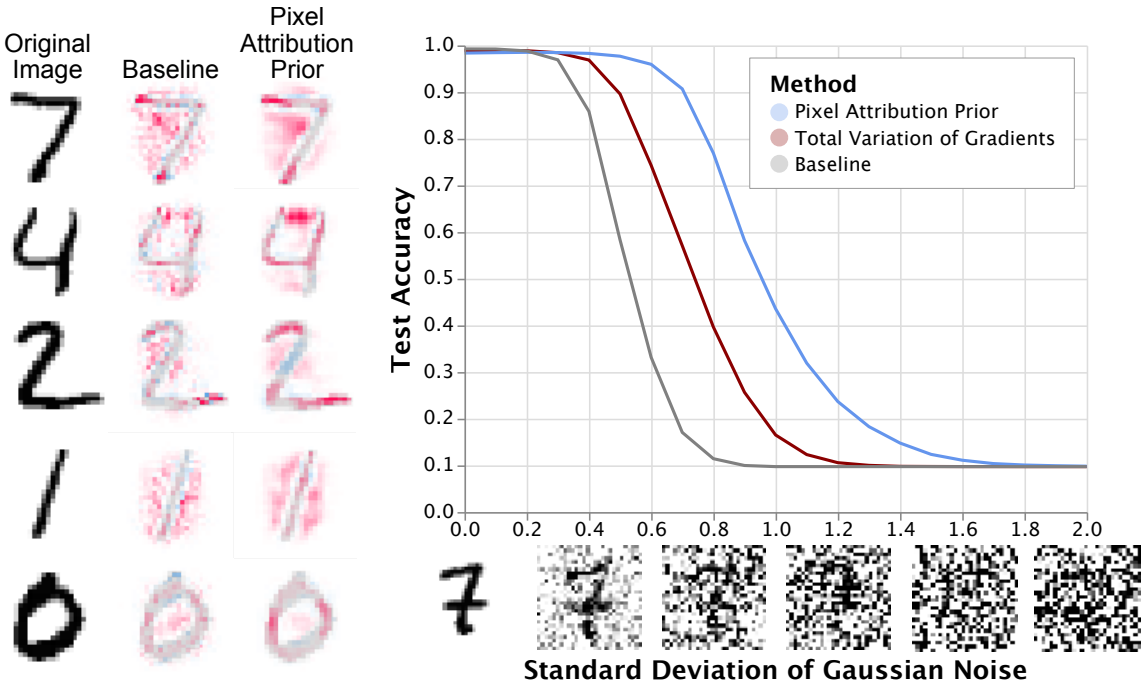


Figure 3.13: Left: Expected gradients attributions (with 100 samples) on MNIST for both the baseline model and the model trained with an attribution prior, for five randomly selected images classified correctly by both the baseline and the regularized model. Red pixels indicate pixels positively influencing the prediction, while blue pixels negatively influence the prediction. Training with an attribution prior generates visually smoother attribution maps and tend to better highlight relevant parts of the image. Right: Training with an attribution prior induces robustness to noise, more so than an equivalent model trained by minimizing the total variation of gradients or an equivalent baseline. The baseline model achieves an accuracy of 0.9925, compared 0.9836 for the pixel attribution prior and 0.9888 for the gradient model.

We trained each model type with 22 values of λ ranging from 10^{-3} to 10^{-4}. We calculated average accuracy and total variation of attributions on the test set for each value of λ. For each model type, we selected the model with the lowest total variation of attributions that still achieved accuracy within 1 percent of the best unregularized model to qualitatively evaluate attributions and quantitatively evaluate robustness. We qualitatively evaluated attributions by displaying attribution maps generated by the attribution method penalized in each model. We focus on Figure 3.14, which displays results for two images from a randomly chosen batch. A larger set of images is shown in Figure 3.15. In both figures, the unregularized model is explained with the same settings as the multi-reference EG model.

The attributions in Figure 3.14 vary widely for a given image. The unregularized model is quite noisy but emphasizes important positive and negative features of each image. The "4" is influenced toward its classification by the positive upper left and middle strokes; it is also influenced by the negative upper middle blank space, which would push the classification toward "9" if filled in. Similarly, the "0" is influenced toward its classification by the left and right edges of its curve; it is also influenced by the negative center, which would push the classification toward "8" if filled in. While these features are visible in the unregularized maps, they are obscured by the noisy quality of the attributions. We seek a method that can reveal such positive and negative features while yielding smooth attribution maps.

All models other than the unregularized model yield smooth attributions when explained with the method that was penalized during training – this should be the case for any reasonably effective training process. However, most explanations suffer from issues that EG avoids. The logit gradients method counterintuitively implies that the bottom-right stroke of the "0" digit makes a "0" label less likely and often yields degenerate explanations clustered in the middle of the image (Figure 3.15). Similarly, the log-probability gradients are smooth but do not yield a clear semantic pattern that explains the predictions. Conversely, the IG and single-reference EG explanations highlight clear, contiguous, intuitively important regions such as the positive pixels on the left of the "4" and the "0". However, they are incapable of highlighting important negative blank spaces. Multi-reference EG is able to provide qualitatively similar explanations and highlights the left side of each digit as important, but also highlights the top center of the "4" and the center of the "0" as important blank spaces that drive classification. Only the multi-reference EG regularization is able to provide smooth explanations that highlight clearly meaningful positive *and* negative pixels.

We evaluated robustness by adding normally distributed random noise to each pixel in the test set and recalculating test performance. Figure 3.16 shows the test accuracy of all models as a function of the standard deviation of noise added (standard deviations range higher than in the maintext due to different standardization of the pixel value range). As expected, the unregularized model performs worst as noise is added. Both gradient methods and IG provide small performance improvements; single- and multi-reference EG provide much larger ones. This provides evidence that much of the improved performance under EG comes from its random interpolation step; single-reference EG involves performing a similar process to IG but with random interpolation. This leads to performance matching that of multi-reference EG. However, only multi-reference EG is capable of providing robustness, smooth attributions, and meaningful highlighting of background pixels.

3.5.8 ImageNet Experiments

In this section, we detail experiments performed on applying Ω_{pixel} to classifiers trained on the ImageNet 2012 challenge [121].

Experimental Setup

We use the VGG16 architecture introduced by [83]. We then fine-tune the network from its pre-trained weights in the Tensorflow Slim package [122]. We fine-tune on the ImageNet 2012 training set using the original cross entropy loss function in addition to Ω_{pixel} using asynchronous gradient updates with a batch size of 16 split across 4 Nvidia 1080 Ti GPUs. During fine-tuning, we use the same training procedure outlined by [122]. This includes randomly cropping training images to 224×224 pixels, randomly flipping images horizontally, and normalizing each image to the same range. To optimize, we use gradient descent with a learning rate of 0.00001 and a momentum of 0.9. We use a weight decay of 0.0005, and set $\lambda = 0.00001$ for the first epoch of fine-tuning, and $\lambda = 0.00002$ for the second epoch of fine-tuning. As with the MNIST

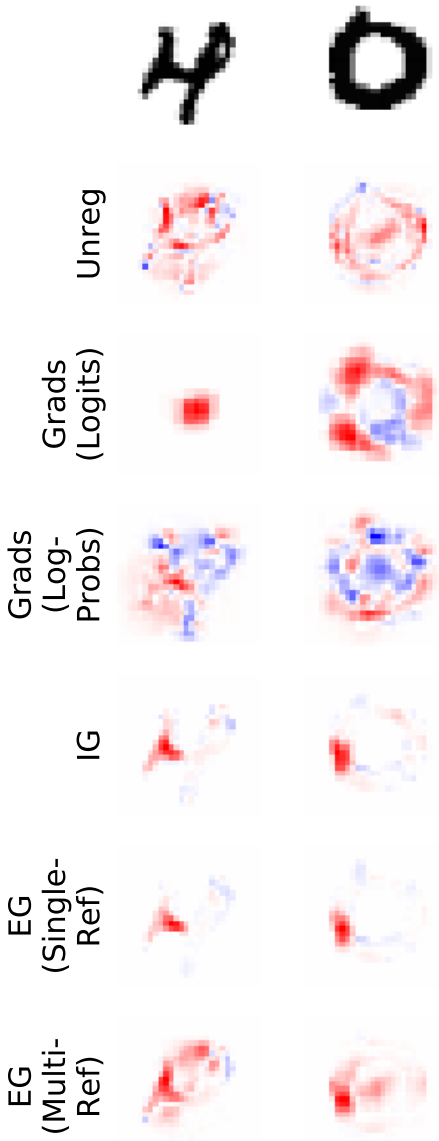


Figure 3.14: Attribution prior model explanations for two samples from a random batch of MNIST data. Rows correspond to each attribution prior method; each method is explained with the same attribution method that was penalized.

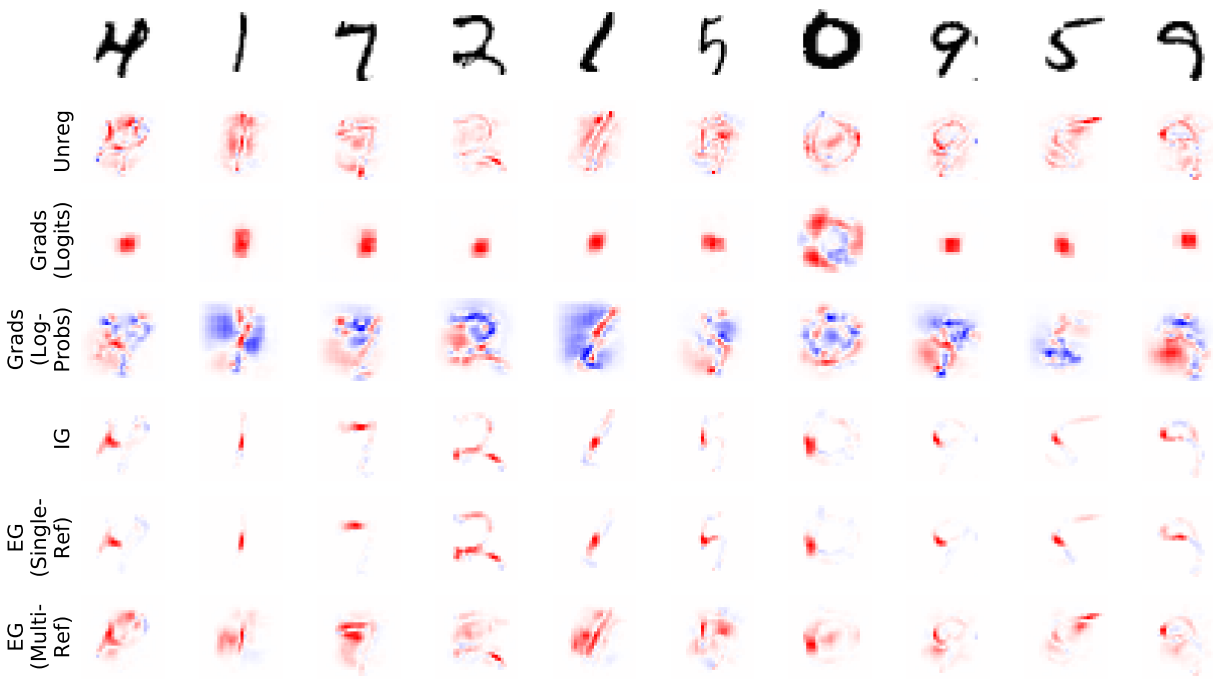


Figure 3.15: Attribution prior model explanations for 10 samples from a random batch of MNIST data. Rows correspond to each attribution prior method; each method is explained with the same attribution method that was penalized.

Table 3.4: Performance of the VGG16 architecture on the ImageNet 2012 validation dataset before and after fine-tuning.

Model	Top 1 Accuracy	Top 5 Accuracy
Baseline	0.709	0.898
Image Attribution Prior 1 Epoch	0.699	0.886
Image Attribution Prior 1.25 Epochs	0.674	0.876

experiments, we normalize the feature attributions before taking total variation.

Results

We plot the attribution maps on images from the validation set using expected gradients for the original VGG16 weights (Baseline), as well as fine-tuned for 320,292 steps (Image Attribution Prior 1 Epoch) and fine-tuned for 382,951 steps, in which the last 60,000 steps were with twice the λ penalty (Image Attribution Prior 1.25 Epochs). Figure 3.17 demonstrates that fine-tuning using our penalty results in sharper and more interpretable image maps than the baseline network. In addition, we also plot the attribution maps generated by two other methods: integrated gradients (Figure 3.18) and raw gradients (Figure 3.19). Networks regularized with our attribution prior show more clear attribution maps using any of the above methods, which implies that the network is actually viewing pixels more smoothly, independent of the attribution method chosen.

We note that in practice, we observe similar trade-offs between test accuracy and interpretability/robustness mentioned in [70]. We show the validation performance of the VGG16 network before and after fine-tuning in Table 3.4 and observe that the validation accuracy does decrease.

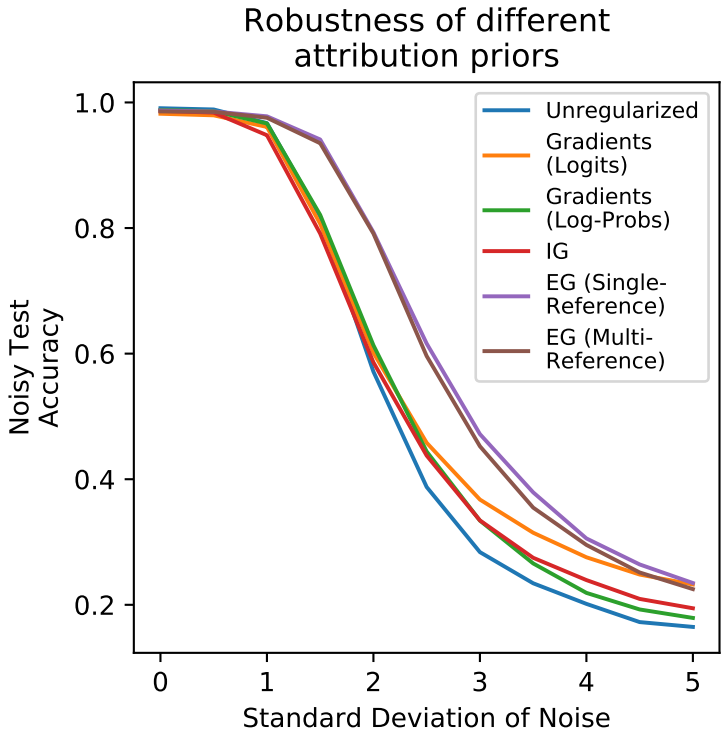


Figure 3.16: Robustness of attribution prior models. Test accuracy of all models decreases as the standard deviation of noise added to the test set increases. Both EG methods are much more robust than the other methods, though IG and gradient-based methods also demonstrate moderate robustness improvements.

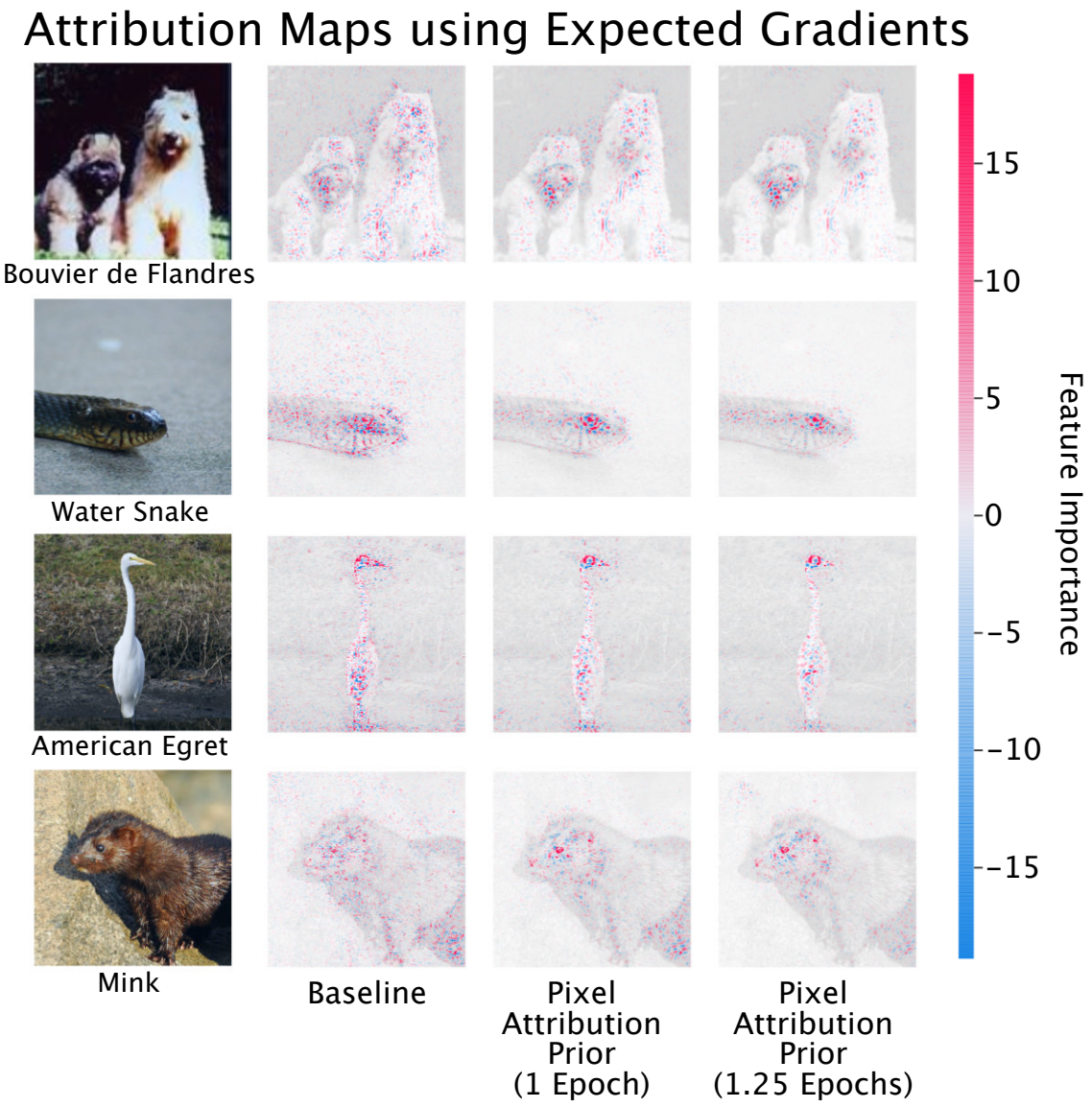


Figure 3.17: Attribution maps generated by Expected Gradients on the VGG16 architecture before and after fine-tuning using an attribution prior.

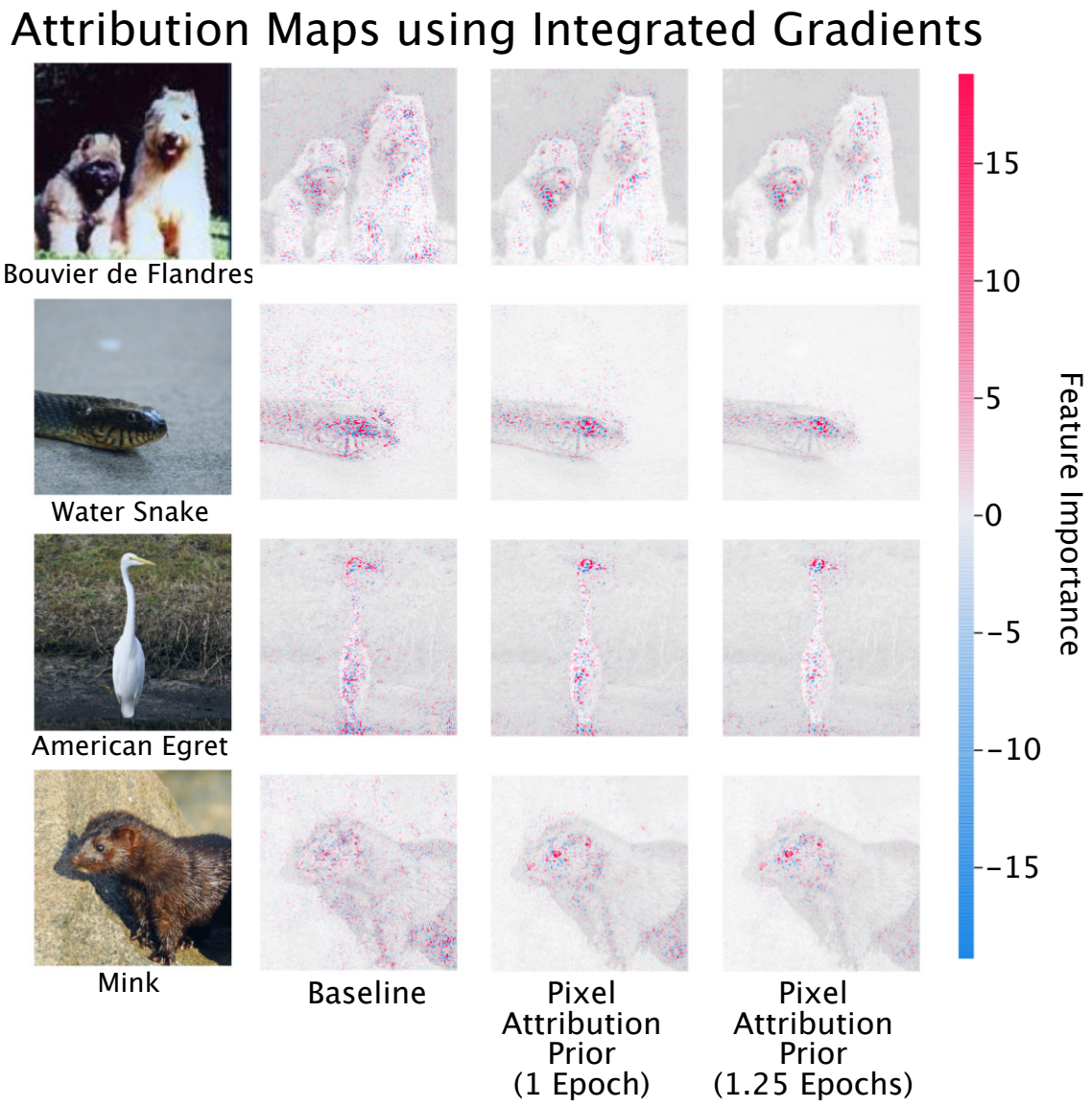


Figure 3.18: Attribution maps generated by Integrated Gradients on the VGG16 architecture before and after fine-tuning using an attribution prior.

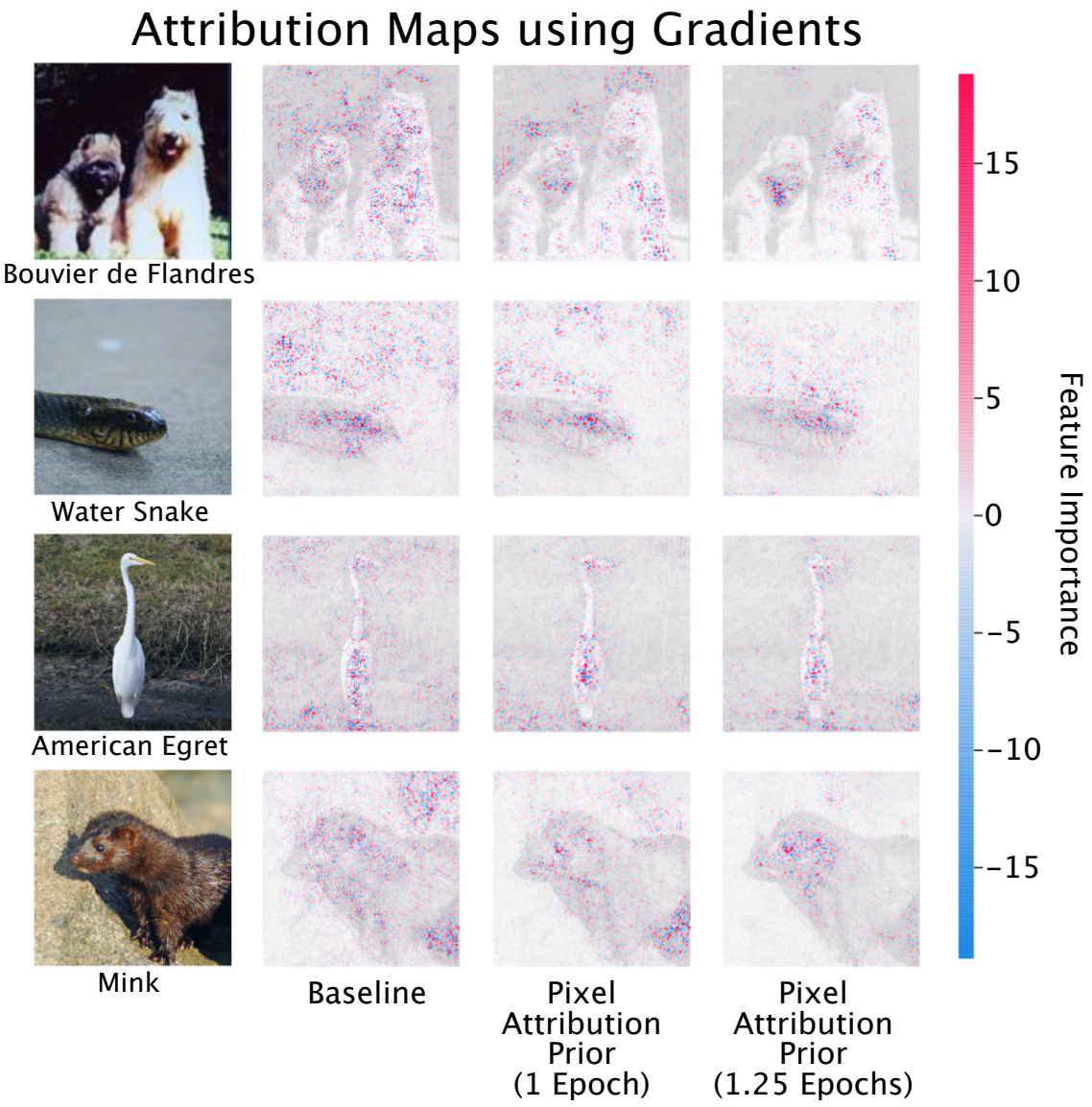


Figure 3.19: Attribution maps generated by raw gradients on the VGG16 architecture before and after fine-tuning using an attribution prior.

3.5.9 Biological experiments

RNA-seq preprocessing

To ensure a quality signal for prediction while removing noise and batch effects, it is necessary to carefully preprocess RNA-seq gene expression data. For the biological data experiments, RNA-seq were preprocessed as follows:

1. First, raw transcript counts were converted to fragments per kilobase of exon model per million mapped reads (FPKM). FPKM is 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 [123]. FPKM is calculated as follows:

$$FPKM = \frac{X_i \times 10^9}{N l_i} \tag{3.4}$$

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

2. Next, we removed non-protein-coding transcripts from the dataset.

3. We removed transcripts that were not meaningfully observed in our dataset by dropping any transcript where $> 70\%$ measurements across all samples were equal to 0.

4. We $\log_2$ transformed the data

5. We standardized each transcript across all samples, such that the mean for the transcript was equal to zero and the variance of the transcript was equal to one:

$$X_i^{'} = \frac{X_i - \mu_i}{\sigma_i} \tag{3.5}$$

   where X_i is the expression for a transcript, μ_i is the mean expression of that transcript, and σ_i is the standard deviation of that transcript across all samples.

6. Finally, we corrected for batch effects in the measurements using the ComBat tool available in the sva R package [124].

Further details on experimental results

Since we added graph-regularization to our model by fine-tuning, we wanted to ensure that the improved performance did not simply come from the additional epochs of training *without* the attribution prior. We use a two-tailed dependent t-test to compare the R^2 attained from 10 independent retrainings of the regular neural network to the R^2 attained from 10 independent retrainings of the neural network with the same number of additional epochs that were optimal when adding the graph penalty (see Figure 3.20). We found no significant difference between the test error of these models ($p = 0.4828002346791498, T = 0.7319884084858485$).

Likewise, in order to see if the improved performance was simply due to additional regularization, we also compared to a *uniform* prior on the attributions – i.e. an L2-norm penalty on the vector of average magnitude expected gradients attributions during training. While the average $R2$ across 10 independent retrainings of the neural network with the uniform prior appeared to be slightly higher than the $R2$ of the standard neural network, this result was not significant according to a two-tailed dependent t-test ($p = 0.12256623182371867, T = 1.7040639451054609$, see Figure 3.20).

To ensure that the models were learning the attribution metric we tried to optimize for, we compared the explanation graph penalty ($\bar{\phi}^T L_G \bar{\phi}$) between the unregularized and regularized models, and found that the graph penalty was on average nearly two orders of magnitude lower in the regularized models (see Figure 3.22). We also examined the pathways that our top attributed genes were enriched for using Gene Set Enrichment Analysis and found that not only did our graph attribution prior model capture far more significant pathways, it also captured far more AML-relevant pathways (see Figure 3.21). We defined AML-relevant by a query for the term "AML," as well as queries for AML-relevant transcription factors.

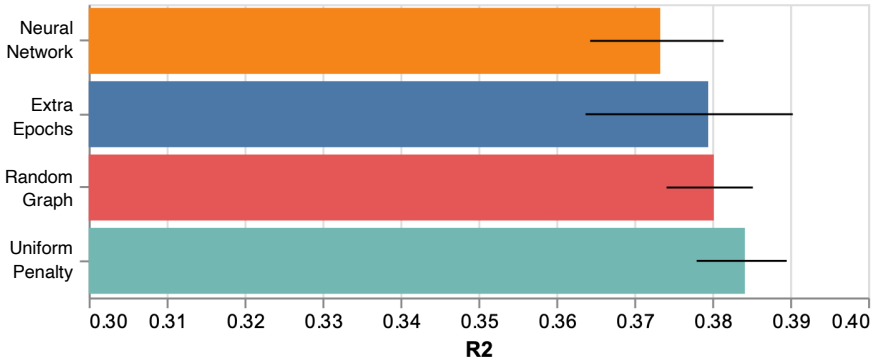


Figure 3.20: Fine tuning *without* graph prior penalty leads to no significant improvement in model performance. Additionally, uniform penalty on L2 norm of global attributions does not significantly increase test set R^2. (Means and error bars calculated by the same method as in the main text).

Most important genes for neural network *with attribution prior* come from biologically-relevant pathways

Pathway	FDR q-value
RNA Pol I Promoter Opening	$< 10^{-280}$
Amyloids	0.002722
Down-regulated in T Lymphocyte and NK Progenitor cells	0.006435
Down-regulated in normal aging	0.007065
TEL pathway	0.007384
B Cell Lymphoma Cluster 7	0.007601
AML Cluster 9	0.007604
Response to MP470 up	0.007853
Upregulated genes in cells immortalized by HOXA9 and MEIS1	0.008068
AML Cluster 12	0.008163

… +145 more pathways

Most important genes for neural network *without attribution prior* are not significantly enriched for any AML-related pathways

Pathway	FDR q-value
RNA Pol I Promoter Opening	0.001778
Amyloids	0.004001

No additional pathways significant after FDR correction

Figure 3.21: Top pathways for neural networks with and without attribution priors

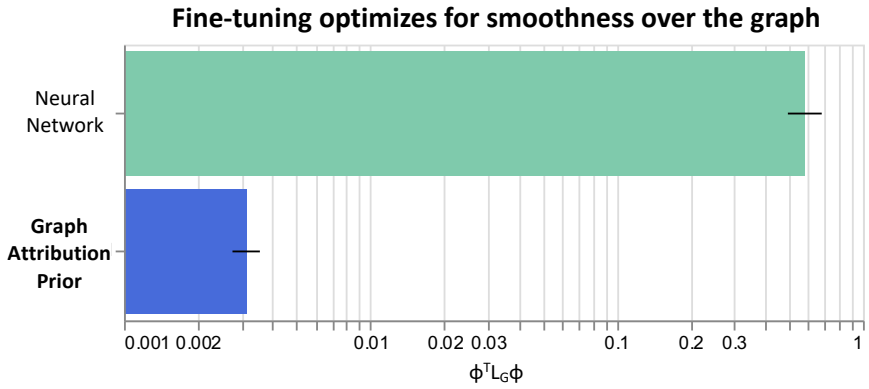


Figure 3.22: Fine tuning optimizes for the metric we care about: smoothness over the graph

3.5.10 Sparsity experiments

Model

Regularizers: We tested a large array of regularizers. See the Methods section and section 3.5.10 for details on how optimal regularization strength was found for each regularizer. *Italicized* entries were displayed in the main text. The optimal regularization strength from validation-set tuning is listed in the table below. Regularizations that involve attributions calculate attributions with respect to the output logits unless otherwise specified.

- *Sparse Attribution Prior* - Ω_{sparse} as defined in the main text.
- Mixed L1/Sparse Attribution Prior - Motivated by the observation that the Gini coefficient is normalized and only penalizes the *relative distribution* of global feature importances, we attempted adding an L1 penalty to ensure the attributions also remain small in an absolute sense. The L1 and Gini terms were equally weighted. This did not result in statistically significant improvements to performance or sparsity (subsection 3.5.10).
- *Sparse Group Lasso* - Rather than simply encouraging the weights of the first-layer matrix to be zero, the sparse group lasso also encourages entire columns of the matrix to shrink together by placing an L2 penalty on each column. As in [96], we added a weighted sum of column-wise L2 norms to the L1 norms of each layer's matrix, without tuning the relative contribution of the two norms (equal weight on both terms). We also follow [96] and penalize the absolute value of the *biases* of each layer as well.
- *Sparse Group Lasso First Layer* - This penalty was similar to [96], but instead of penalizing *all* weights and biases, only the first-layer weight matrix was penalized. This model outperformed the SGL implementation adapted from [96], but did not outperform the sparse attribution prior.
- L1 First-Layer - In order to facilitate sparsity, we placed an L1 penalty on the input layer of the network. No regularization was placed on subsequent layers.
- *L1 All Layers* - This penalty places an L1 penalty on all matrix multiplies in the network (not just the first layer).
- L1 Expected Gradients - This penalty penalizes the L1 norm of the vector of global feature attributions, $\bar{\phi}_i$ (analogous to how LASSO penalizes the weight vector in linear regression).

- L2 First-Layer - This penalty places an L2 penalty on the input layer of the network, with no regularization on subsequent layers.
- L2 All Layers - This penalty places an L2 penalty on all matrix multiplies in the network (not just the first layer).
- L2 Expected Gradients - This penalty penalizes the L2 norm of the vector of global feature attributions, $\bar{\phi}_i$ (analogous to how ridge regression penalizes the weight vector in linear models).
- Dropout - This penalty "drops out" a fraction p of nodes during training, but uses all nodes at test time.
- *Baseline (Unregularized)* - Our baseline model used no regularization.
- *L1 Gradients (Log-Probabilities, Ross et al. 2017)* - To achieve the closest match to work by [68, 97], we placed a L1 penalty on the global gradients attribution vector of the network (mean across all samples of the absolute value of the gradient for each feature). This is similar to the "neural LASSO" of [97], but with a goal of global sparsity (a model that uses few features overall) rather than local sparsity (a model that uses a small number of possibly different features for each sample). Whereas our other, non-gradient-based, attribution priors calculated attributions with respect to the model's logits, this penalty calculated attributions with respect to the sum of the log-probabilities assigned to each class in order to more closely match the methodology in [68, 97].
- L1 Gradients (Logits) - This method penalizes the L1 norm of the vector of global gradients as in the Ross et al. penalty [68, 97] but uses the logits instead of the sum of output log-probabilities to more closely match the sparse attribution prior.
- *Gini Gradients (Logits)* - An intermediate step between [68, 97] and our sparse attribution prior would use gradients as an attribution, but our Gini coefficient-based sparsity metric as a penalty. In this model we encouraged a large Gini coefficient of the mean absolute value of the gradients attributions of the model, averaged over all samples. Like the sparse attribution prior, this penalty uses attributions with respect to the model's output logits.
- Gini Gradients (Log-probabilities) - This is identical to the Gini gradients (logits) penalty, but for comparison with the Ross et al. 2017 uses attributions with respect to the sum of the model's output log-probabilities.

The maintext figures, with experiments repeated 200 times, compared the sparse attribution prior to methods previously used in literature on sparsity in deep networks – the L1 penalty on all layers, the sparse group lasso methods [96], and the L1 gradients (log-probabilites) penalty [97]. We also evaluated the Gini gradients (logits) penalty in these experiments because it exhibited the better performance than the Gini gradients (log-probabilities) penalty. The other methods were evaluated similarly but not displayed in the maintext for space reasons and because there was less literature support. The optimal regularization parameters for the methods evaluated in the maintext are as follows (median over all replicates):

	Lambda
Sparse Attribution Prior	1.259e+00
Mixed L1/Sparse Attribution Prior	1.259e+00
L1 (All Parameters)	3.162e+02
L2 (All Parameters)	6.310e-01
SGL (All Parameters)	1.000e-03
SGL (First Layer)	3.981e+00
L1 (First Layer)	1.000e+04
L2 (First Layer)	1.790e+03
L1 (Expected Gradients)	5.012e+00
L2 (Expected Gradients)	1.000e-02
Dropout	6.598e-01
L1 (Gradients, Logits)	1.995e+00
Gini (Gradients, Logits)	1.259e+00
L1 (Gradients, Log-Probs - Ross '17)	1.000e+01
Gini (Gradients, Log-Probs)	1.000e+00
Unregularized	N/A

Hyperparameter tuning:

Most parameter tuning is discussed in the maintext and Methods sections. Dropout required special tuning; we tuned the dropout probability with 121 points linearly spaced over $(0, 1]$. Mean validation performance and sparsity across the range of tuning parameters is shown in Figure 3.23.

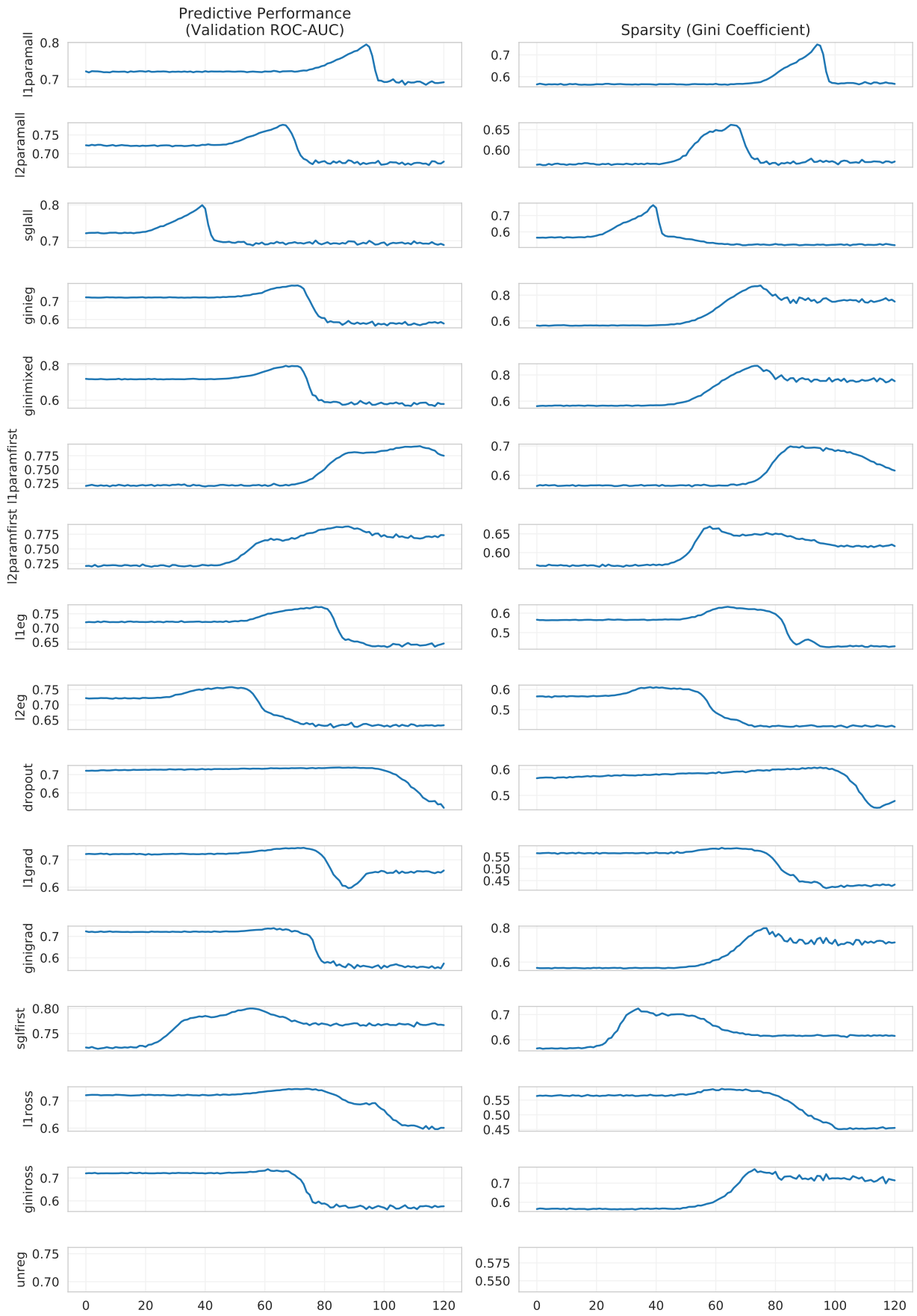


Figure 3.23: Validation performance and Gini coefficient as a function of regularization strength for all models, averaged over 200 subsampled datasets.

Additional Results (Maintext Experiments)

Statistical significance: Statistical significance was assessed as discussed in the Methods section. The resulting p-values and test statistics T for the performance difference between the sparse attribution prior and the other maintext methods were:

	ROC-AUC (p)	ROC-AUC (T)	Sparsity (p)	Sparsity (T)
L1 (All Parameters)	2.523e-05	4.314e+00	1.188e-37	1.602e+01
L2 (All Parameters)	1.251e-28	1.308e+01	4.477e-78	3.098e+01
SGL (All Parameters)	1.400e-07	5.461e+00	1.540e-33	1.468e+01
Mixed L1/Sparse Attribution Prior	7.697e-01	-2.932e-01	6.169e-01	5.010e-01
L1 (First Layer)	5.209e-07	5.188e+00	4.627e-65	2.567e+01
L2 (First Layer)	6.979e-17	9.148e+00	5.446e-88	3.548e+01
L1 (Expected Gradients)	7.143e-24	1.152e+01	7.315e-85	3.401e+01
L2 (Expected Gradients)	1.926e-39	1.661e+01	6.557e-88	3.544e+01
Dropout	5.555e-41	1.712e+01	2.667e-100	4.170e+01
L1 (Gradients, Logits)	1.459e-44	1.831e+01	3.923e-97	4.002e+01
Gini (Gradients, Logits)	7.793e-35	1.510e+01	3.612e-27	1.261e+01
SGL (First Layer)	4.571e-14	8.128e+00	3.667e-79	3.146e+01
L1 (Gradients, Log-Probs - Ross '17)	6.834e-46	1.876e+01	7.602e-102	4.253e+01
Gini (Gradients, Log-Probs)	2.518e-35	1.526e+01	4.089e-32	1.422e+01
Unregularized	9.821e-56	2.220e+01	1.246e-107	4.579e+01

Feature Importance Summary: We also show summaries of the mean absolute feature importance for the top 20 features in each model in Figure 3.24.

Performance and Sparsity: We also display test ROC-AUC and sparsity results (with means and error bars calculated by the same method as in the maintext) for all possible penalties in Figure 3.25.

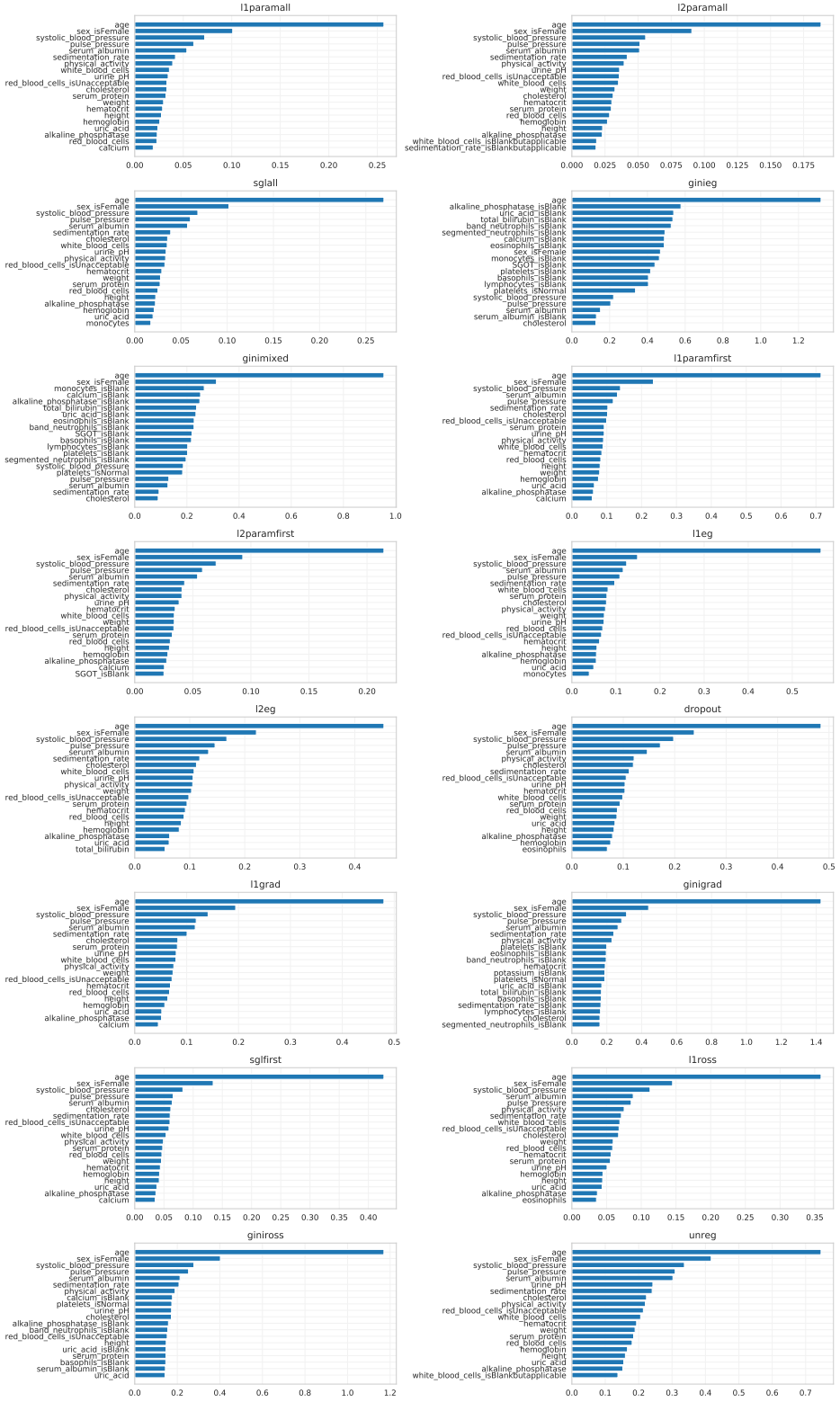


Figure 3.24: Summary of feature attributions for top 20 features from each model (best model from each class chosen as described in the main text); importances are an average over the 200 small-data subsamples.

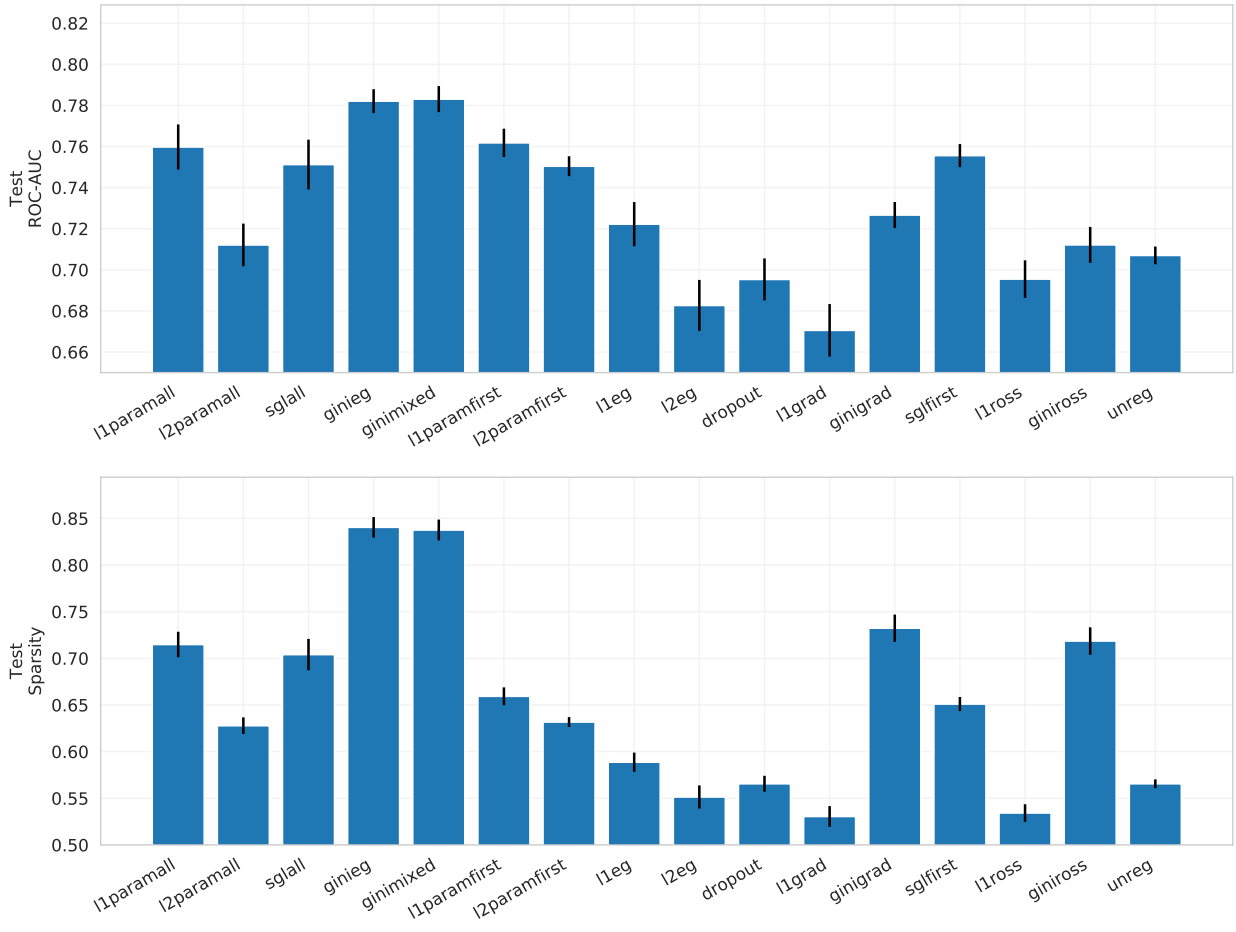


Figure 3.25: Performance and sparsity bar plots for all models.

Chapter 4

Decoupled Regression: Meta-analytic synthesis of multivariate predictive models

The field of meta-analysis focuses on synthesizing knowledge from separate studies to improve estimates of important scientific information. However, meta-analysis of predictive models, i.e., combining models from several studies into a single predictive model, has yet to receive full attention. In medicine, such meta-analyses could speed development of accurate clinical risk scores for new clinical outcomes or upgrade the quality of predictive models when data sharing is difficult or impossible. We examine meta-analysis of generalized linear models (GLMs) from studies with non-overlapping features. Several current approaches to this problem suffer from major drawbacks. First, non-overlapping features cause the naive approach of averaging reported coefficients to exhibit substantial bias. Second, while more sophisticated methods exist, we are aware of no such methods that support arbitrary GLM link functions. Finally, all GLM meta-analysis methods require the user to have access to additional data beyond the observed studies; the least burdensome requirement among existing methods is access to a labeled dataset. To overcome these problems, we introduce a method called *decoupled regression*, which provides exact solutions for synthesis of any set of GLMs with canonical link functions. Decoupled regression requires only unlabeled, not labeled, covariate data, and it naturally supports regularization and model selection to handle problems like small sample size and covariate shift. Our code is available as simple open-source software; we believe it will facilitate creation of higher quality predictive models in both clinical medicine and a wide array of scientific fields.

4.1 Main

Meta-analysis describes a large body of research methods that synthesize information from separate studies to accurately estimate important scientific parameters. Physicians place particular value on these methods as a way to provide high-quality information for high-stakes decisions and to summarize a sizable literature base for providers who are often fully occupied with patient care responsibilities. Most meta-analyses typically estimate a single effect, such as "How much does a particular drug reduce mortality in patients at high risk for heart attacks?", by combining estimates from several relevant studies. A less-studied question is how to create multivariate predictive models from published data; for example: "Study A predicts lung disease using age, sex, and blood pressure, and Study B predicts lung disease using white blood cell count. How can I predict lung disease using all four variables?"

This paper contributes a meta-analysis method for generalized linear models (GLMs) called *decoupled regression*. Given an *unlabeled dataset* containing m features, we show how to synthesize a generalized model from any set of published GLM coefficients on arbitrary subsets of the m features. This encompasses cases where each study is a simple association (a univariate GLM), a multivariate GLM, or a mixture of the two cases.

This kind of analysis has value in many clinical scenarios. For example, during the early months of the COVID-19 pandemic, univariate mortality odds-ratios were published for patient demographics, disease history, and laboratory values, but it took several more months to create multivariate risk scores [125, 126, 127, 128]. The use of decoupled regression could have enabled much earlier creation of multivariate clinical risk scores for COVID-19 mortality and facilitated more accurate early triage. A second example is medical review articles, which often publish a list of univariate odds-ratios that support or reject a particular diagnosis (i.e., diabetes). However, naively combining these odds-ratios leads to biased risk estimates. The use of decoupled regression could combine any set of odds-ratios to provide a more accurate risk estimate (clinical risk score). Overall, effective meta-analysis of multivariate models is valuable in any area where meta-analysis is used, particularly when studying emerging diseases or when data cannot easily be shared.

One naive solution to multivariate meta-analysis is to average the coefficients of each published model. When each study is univariate, this involves simply concatenating each association. However, because features like age and blood pressure are correlated, univariate associations between age and mortality will represent the effect of *both* age and blood pressure on mortality (and likewise for blood pressure). In this case, naively combining the univariate effects of age and blood pressure will overestimate a patient's risk. In contrast, decoupled regression combines coefficients while accounting for correlation.

Several early methods synthesize linear and logistic regression models under the assumption that each model was trained on the same set of covariates [129, 130]. Instead, we address the more challenging task of synthesizing models trained on *different* sets of covariates; the resulting coefficients are conditioned on different variables and not directly comparable. Some methods address this by incorporating additional

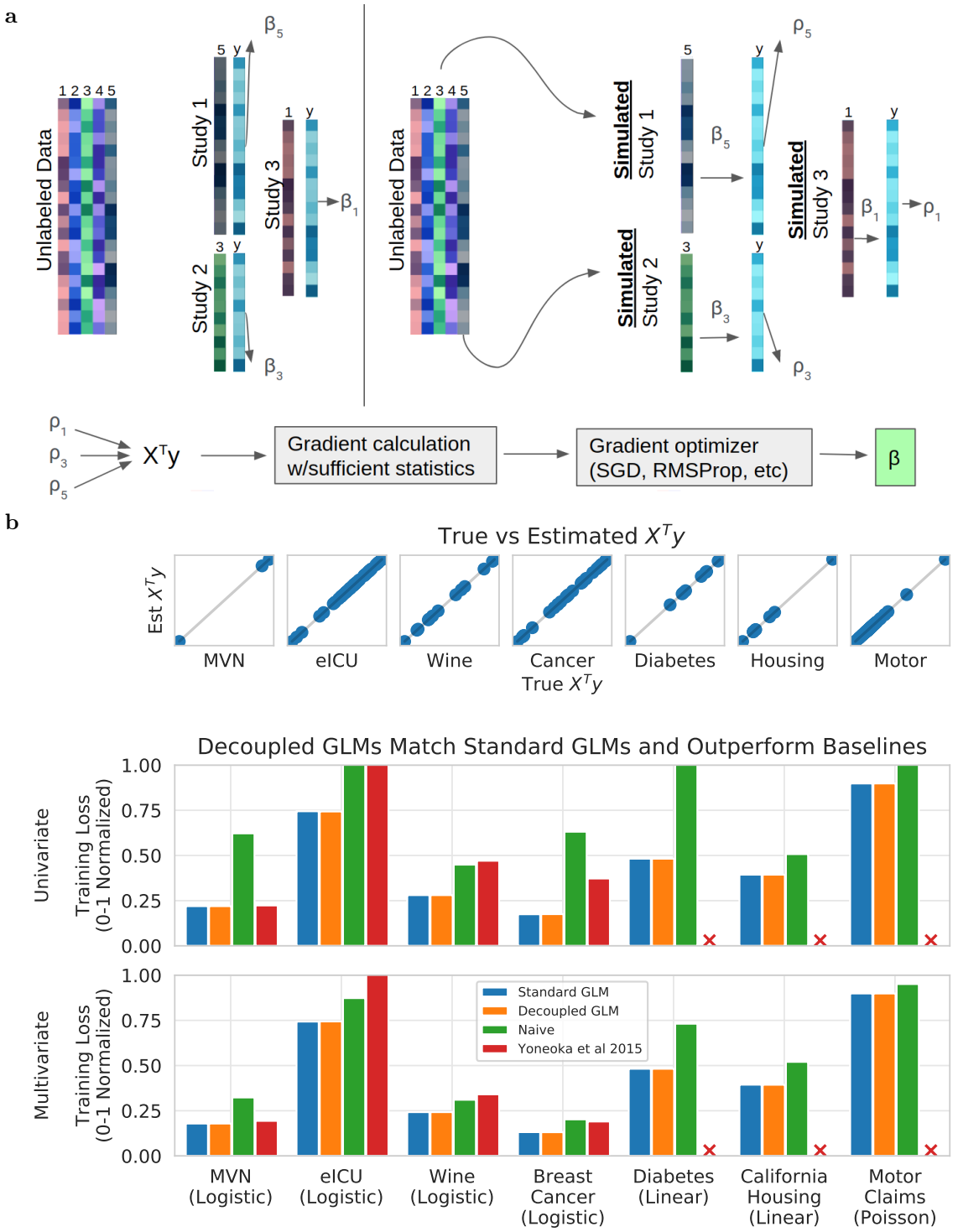


Figure 4.1: Overview and equivalence of decoupled and standard GLMs. a) Conceptual overview of decoupled regression. Assume we have unlabeled data but observe regression coefficients reported by studies with access to labeled data. We can simulate the studies using the observed regression coefficients β_i in our own data and calculate $\rho = X^T y$ from the resulting estimates of y. This lets us calculate the gradients of a GLM and solve for its parameters. b) A benchmark confirming that decoupled GLMs match standard GLMs trained on labeled data and outperform other methods. (first row) Confirmation that we can correctly recover $X^T y$ for all 7 datasets. (second row) Performance when synthesizing univariate associations. (third row) Performance when synthesizing multivariate models. X indicates the model was not applicable to a task.

information from each study, such as covariance matrices or entire datasets [131, 132, 133]. Others have attempted to infer study covariance matrices from the reported regression coefficients alone, but these methods have been shown to work only for very small numbers of features [134].

To our knowledge, the method requiring the least additional information uses a separate, labeled patient dataset to estimate bias correction terms and covariances among coefficients; we refer to this method throughout as IPD (Individual Patient Data) [135]. IPD provides explicit solutions for logistic regression using an approximation that assumes multivariate normal covariates. In contrast, our decoupled regression framework yields a simple, exact solution for any GLM with a canonical link using unlabeled instead of labeled data.

4.2 Results

4.2.1 Decoupled regression is a method for label-free meta-analysis

Our decoupled regression meta-analytic framework starts from the following assumptions:

1. We have access to *unlabeled* training data X_{train} containing m features.
2. We have access to component models $f_1 ... f_k$, with weights $\beta_1 ... \beta_k$, which are maximum-likelihood GLMs with canonical link functions.
3. Each component model is trained on a subset S_i of the m features. These subsets may be distinct or overlapping across the component models.
4. The distribution of patient features and outcomes X, y is the same for X_{train} and the data for each component on which the model was trained.

Assumption 4 is similar to the [135] assumption that "the distribution of covariates and outcomes is common across the studies in the meta-analysis."

For GLMs with a canonical link, it is well known that the vector $X^T y$ is sufficient to estimate the parameter vector β [136]. A key component of decoupled regression is the following expression for the gradient of the log-likelihood for any GLM with a canonical link:

$$\frac{\partial}{\partial \beta} \ell(\theta) \propto (X^T - X^T \hat{y}), \tag{4.1}$$

where ℓ is the log-likelihood, X and y are the data and labels, and $\hat{y}$ is the model's predictions with parameters β (detailed derivation in Methods 4.4.2). Equation 4.1 lets us calculate our estimate $\hat{\beta}$ of the synthesized GLM parameters using gradient descent, which requires only X_{train} and $X^T y$. $X^T y$ is proportional to the Pearson correlation between each feature and the outcome; if we can estimate it from published studies, we can solve the GLM.

Unfortunately, $X^T y$ is usually not reported unless a study focuses on Pearson correlations (i.e., univariate studies with continuous outcomes). While a rich meta-analysis literature discusses how to convert between different types of associations in order to compare effect sizes, these conversions are not perfect; it is impossible to perfectly translate β_i into $X^T y$ in general without additional information. We leverage our unlabeled data X_{train} to do so in a statistically consistent way: for study i, we let X_i be the matrix formed by the columns of X_{train} contained in study i. We then define $\hat{y}_i = f_i(X_i)$. Our estimate of $X^T y$ in study i, denoted as $X^T y_i$, is:

$$X^T y_i = X_i^T \hat{y}. \tag{4.2}$$

Under assumption 4 and the optimality condition implied by equation 4.1 (Methods 4.4.2), this exactly recovers ρ_i. After taking a weighted average across studies, we can use Equation A to train a GLM (Figure 4.1A).

Table 4.1: Overview of 7 benchmark datasets.

Datasets	# Samples	# Features	Outcome	Regression Type	Base Rate
MVN	1000	3	Synthetic	Logistic	0.289
eICU	11501	42	Mortality	Logistic	0.107904
Wine	178	13	Wine Type	Logistic	0.606742
Breast Cancer	569	30	Malignant/Benign	Logistic	0.374341
Diabetes	442	10	Disease Progression	Linear	152.133
California	20640	8	House Value	Linear	2.06856
Motor	678013	75	Claim Amount Per Policy Interval	Poisson	0.263964

4.2.2 Access to univariate associations recovers correct GLMs

Our first goal was to empirically test whether we could correctly train GLMs using only unlabeled data and regression coefficients. We constructed a benchmark to evaluate the performance of decoupled regression across many datasets and model types, including 4 datasets using logistic regression models, 2 datasets with least-squares linear regression models, and 1 with a Poisson regression model (Table 4.1 and Figure 4.1b, Methods 4.4.1). In each dataset, we hid the labels and learned a multivariate predictive model given only the unlabeled data and two types of simulated "study coefficients." In the first "univariate" experiment, available study coefficients consisted of each univariate odds-ratio in the training set. In the second "multivariate" experiment, study coefficients consisted of two GLMs, each built on half of the full set of features.

We first verified that our estimator correctly recovered $X^T y$ in all datasets in the univariate experiment in Figure 4.1b (results for the multivariate experiment were identical). Next, we evaluated whether several methods could achieve a training error (log-loss, mean-squared error, and Poisson deviance) equivalent to that of a GLM trained with labeled data. Due to the wide array of prediction tasks, we normalized error in each case so that perfect predictions received a normalized error of 0 and predicting the mean received a normalized error of 1.

Across the range of datasets and prediction tasks, decoupled regression GLMs (with access to β_i) performed almost identically to traditional GLMs (with access to y_i) (Figure 4.1B) [15]. Further, decoupled GLMs always achieved a lower training error than two competing methods: the naive averaging approach and the IPD method ([135]). Together, these results establish that decoupled regression successfully combines regression coefficients to recover GLMs in a wide range of modeling situations.

4.2.3 Regularization provides robustness to study biases

Our benchmark experiments show that decoupled regression was effective in a scenario where its assumptions were met. Specifically, because all "study" coefficients were calculated on the training set, the distributions of training and study data were indeed identical (Assumption 4). In real-world meta-analysis, Assumption 4 is usually violated; each study has site-specific biases in data distribution and resulting coefficients.

We hypothesized that this bias would increase errors in the decoupled regression model. We tested this hypothesis by training new decoupled GLMs on eICU data. As in the univariate benchmark, we estimated univariate coefficients β_i in the same dataset $(X_{\text{train}}, y_{\text{train}})$ as the unlabeled data. However, we also added a random bias to each β_i. When we evaluated performance on held-out test data (averaged over 3 random noisy replicates), standard decoupled regression (Figure 4.2a, blue line) suffered from very low performance, even underperforming naive averaging when sufficient bias was added to the study coefficients.

We further hypothesized that this problem resulted from the unregularized decoupled GLM overfitting to biased studies. We tested this by retraining our decoupled GLM with an L1 penalty. We found that strong regularization ($\lambda = 1000$) improved performance and that selecting the optimal λ (found in this case by tuning on the test set) resulted in minimal error even at a relatively high bias. However, we still needed to choose the correct regularization parameter. Usually, this would involve cross-validation, but it was not straightforward to calculate validation loss because we did not assume access to labeled data.

Information criteria like AIC and BIC are sometimes used to perform model selection when cross-validation

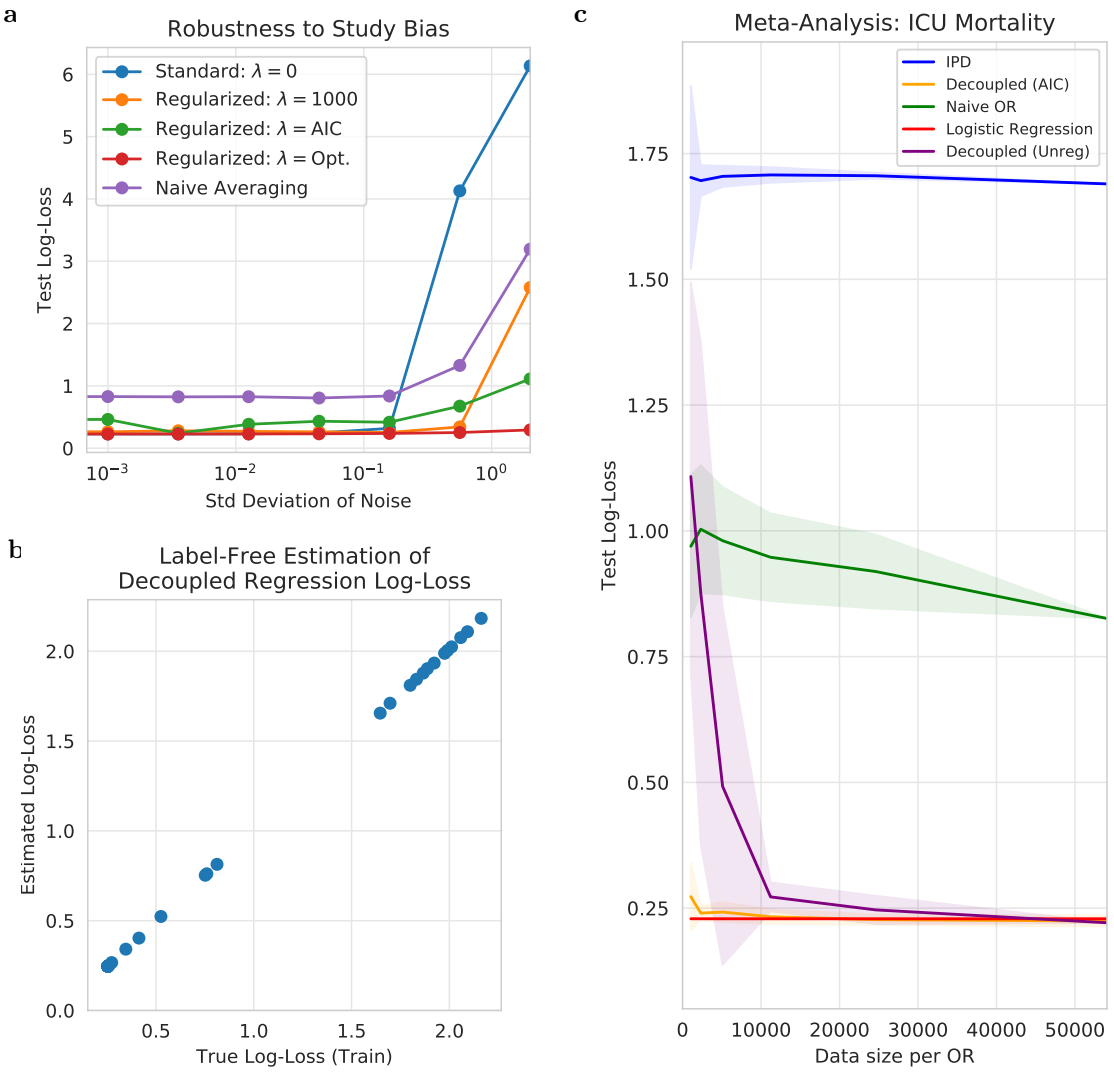


Figure 4.2: Meta-analysis of decoupled GLMs. a) As bias is added to odds-ratios, unregularized models' losses increase. Proper regularization improves performance. b) Our label-free log-loss estimate for decoupled models closely corresponds to the true log-loss on the eICU dataset from the univariate benchmark for models with widely varied regularization strengths. c) Performance on simulated meta-analysis as a function of dataset size per odds-ratio. Both decoupled regression models significantly outperform both the IPD method and naive averaging, but regularization tuned with AIC greatly reduces the amount of data required to achieve low error. Shading represents 95 percent confidence intervals over 25 different random study sets.

Table 4.2: Detail on eICU geographic regions.

eICU Regions	Geographic Region	# Samples	# Hospitals	Base Rate
Train	Northeast	11501	13	0.107904
Study	South	54080	52	0.0930288
Test	West	34212	43	0.0762306

is difficult. Both require access to each candidate model's log-likelihood and degrees of freedom. With an L1 penalized model, the degrees of freedom can be estimated as the number of nonzero coefficients [137]. Calculating log-likelihood is more complicated because it usually requires labels, which we do not assume access to. However, we show that log-likelihood can be estimated by integrating the gradients given in Equation 4.1 (Figure 4.2b, detailed derivation in Methods 4.4.5). In Figure 4.2b, we trained L1-regularized GLMs with 51 values of λ logarithmically spaced in $[1e-10, 1e10]$ on the univariate benchmark eICU data. We found that label-free log-loss estimation closely matched the log-loss calculated on labeled data. In our bias experiments, the resulting AIC-based choice of λ never performed worse than naive averaging and suffered from much less error at high levels of bias (Figure 4.2a).

4.2.4 Decoupled GLMs are effective for meta-analysis

We performed a meta-analysis to evaluate generalization performance and robustness to study bias (violations of Assumption 4) in the eICU dataset [47]. Notably, this dataset combines data from patient stays at over 200 intensive care units (ICUs) in distinct geographic regions of the United States (Table 4.2). Our evaluation aimed to use 42 features to predict a binary indicator of inpatient mortality, an important subject of prior research [47, 27, 28]. Our goal was not to outperform previously published ICU mortality models but to demonstrate that a high-quality model can be obtained through meta-analysis, even where a natural domain shift occurs between geographic regions and health systems.

We compared decoupled regression models against naive averaging and IPD. We selected the unlabeled dataset X_{train} from the northeastern United States, calculated univariate mortality odds-ratios β_i for each feature in patient subsets sampled by hospitals from the southern United States, and calculated loss in a test dataset from the western United States (Methods 4.4.4). This scenario introduced a realistic but challenging hospital-specific bias in estimation of each study's odds-ratio.

Figure 4.2c shows test performance for each model as a function of the amount of data (number of patient stays) used to estimate each odds-ratio. As expected from Section 4.2.3, unregularized decoupled regression took hundreds of patients per odds-ratio simply to outperform the naive approach, and tens of thousands to approach the performance of a labeled-data GLM. Error for the naive averaging and IPD methods remained high even as dataset size increased. In contrast, the decoupled regression model with an AIC-selected penalty performed well (Figure 4.2c): it never under-performed naive averaging and remained close in performance to the "best-case" model, i.e., the traditional GLM-trained model with a labeled training set.

4.3 Discussion

We presented decoupled regression, a new method for meta-analysis of generalized linear models that lets users synthesize known regression coefficients from the literature and build a single multivariate model using only unlabeled data. Our work improves on existing methods by: supporting study coefficients with non-overlapping feature subsets; providing exact solutions for any exponential family GLMs with canonical links; not requiring that users provide labels; and supporting regularization and hyperparameter tuning to address covariate shift across studies. We released all code for the project as open-source software.

Our work addresses problems similar to [135], but with several major differences. [135] targets users with labeled data who can train a predictive model on that data alone but want to improve regression coefficient estimates by incorporating additional data. Hence, that research focuses more on error of coefficient estimates than on prediction error. In contrast, we target users who do not have labeled data and may be unable to train any model on their own, but who would still like to create an accurate predictive model. Our analysis

also puts more emphasis on the predictive value of the resulting tool. Both approaches have merit, and future work that assesses both methods' ability to recover the true data generating model would be valuable. Another useful future direction is expanding multivariate meta-analysis methods to model classes beyond GLMs.

We intend for our work to contribute to a future where a wide array of predictive models can be quickly and easily combined to provide precise, state-of-the-art scientific predictive knowledge to all.

4.4 Methods

4.4.1 Datasets

We benchmarked decoupled regression on a seven different datasets, overviewed in Figure 4.1. Further details follow.

- Our multivariate normal (MVN) dataset was a synthetic dataset designed to satisfy the IPD method's assumption of covariates distributed MVN and serve as a best-case scenario for the method [135]. It consisted of three features simulated from a multivariate normal distribution with mean μ and variance Σ

$$\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  . The outcome was generated as a simple binary indicator of whether feature 0 was greater than 0.5. We generated 1000 samples.

- The Philips eICU dataset contains over 43 features describing over 200,000 patient stays. The goal is to predict a binary mortality outcome (logistic regression). Uniquely, the data can be split into four geographic regions or over 200 hospitals (see Methods 4.4.4 for detail).

- The UCI Wine dataset was adapted for logistic regression; the labels were binarized so that a 1 label indicated the most common class (wine type), and a 0 indicated the other classes. The dataset contains 13 features describing 178 samples.

- The UCI Breast Cancer dataset was used for logistic regression; it contains 30 high-level features describing 569 images of fine needle aspirates of breast masses. The goal is to classify each mass as benign or malignant.

- The UCI Diabetes dataset was used for least squares linear regression. The dataset contains age, sex, BMI, blood pressure, and several lab measurements (10 total features) for each of 442 diabetes patients. The goal is to predict a continuous-valued measure of disease progression after 1 year.

- The California Housing dataset contains 20,640 census block districts in California. The goal is to use 8 features describing each area to predict the continuous-valued median house value (least squares linear regression).

- The French Motor Claims dataset contains 75 features describing 678,013 motor vehicle insurance policies. The goal is to predict the rate of claims per year (Poisson regression).

Additionally, we noticed that the IPD method failed when datasets were perfectly separable. This would be expected since it does not use regularization. We avoided this issue by flipping small numbers of labels in the MVN (1/100 labels), Wine (1/30 labels), and Breast Cancer (1/100 labels) datasets to introduce noise.

4.4.2 Decoupled Regression Solution

Given the assumptions in the main text, we show how to recover the optimal weights w^*. Recall that a GLM predicts the mean of an exponential family distribution; that is, a distribution with parameters θ, ϕ and the following PDF:

$$f(y|\theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right),$$

where θ, ϕ are the location and scale parameters, respectively, and the particular choice of the functions a, b, c determines the exact distribution [136]. For example, specific choices of a, b, c can give the Normal, Binomial, Poisson, Gamma, and Inverse Gamma distributions, among others. Additionally, by differentiating the preceding PDF and taking expectations with respect to Y, it can be shown that $b'(\theta) = \mu$, where μ is the mean of the distribution. Thus, if we have estimated some parameters θ for a particular exponential family distribution of interest, the function $b'(\theta)$ yields an estimate of the distribution's mean. In GLMs, this

will be the prediction $\hat{y}$ made by our model. It is true only for the Normal distribution that the location parameter is the mean $\theta = \mu$; for any other exponential family distribution, we must transform μ to model it linearly. We model μ with a *link function* g so that $g(\mu) = x^T\beta$. *Canonical* link functions in a GLM make the simple and natural choice of setting g to be the function describing the relationship between the mean and location parameters: $g(\mu) = \theta = X^T\beta$. This allows finding the optimal parameters β for any GLM. We start with the log-likelihood $\ell(\theta)$ and differentiate with respect to the weights β:

$$\begin{aligned}\frac{d}{d\beta}\ell(\theta) &= \frac{d}{d\beta}\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right) \\ &= \frac{1}{a(\phi)}\left(\frac{d}{d\beta}y\theta - \frac{d}{d\beta}b(\theta)\right) \\ &= \frac{1}{a(\phi)}\left(X^Ty - X^T\hat{y}\right).\end{aligned}$$

In the last step, we use the fact that the GLM has a canonical link so $\theta = x^T\beta$ as well as that $b'(\theta)$ gives the model's output $\hat{y} = \hat{\mu}$. This access to the GLM's gradients with respect to the parameters β (up to a factor of $\frac{1}{a(\phi)}$, which is a constant with respect to β) lets us perform gradient descent to find the optimal β; at optimality we will have

$$X^Ty = X^T\hat{y}.$$

This is a consequence of the *sufficiency* property of GLMs with canonical links; X^Ty is known to be a sufficient statistic for β in any such GLM [136]. The gradient formulas give us an easy way to find the optimal β given access only to X^Ty, which is a summary statistic potentially obtainable from the literature. Under the main text's assumption (4), if X^Ty is consistent across datasets, our meta-analysis will recover the true β.

4.4.3 Correctness Benchmark

In each dataset described in Methods 4.4.1, we performed two types of analysis. First, we simulated a set of m univariate studies, one for each feature. Study i reported a univariate association β_i (as well as an intercept term) between feature i in the training data X^{train} and the outcome y^{train}. Second, we simulated a set of 2 multivariate studies, each containing a random $\frac{m}{2}$ features. Unlike in the first case, in this case the feature set assigned to each study was random, so we performed the analysis 3 times with 3 different random partitions of the features and averaged the results. In both cases, we trained several models on the datasets and compared their training losses.

- **Labeled-Data GLM.** We implemented standard GLMs using a Scipy gradient optimizer to represent optimal performance. We verified that these methods were a close match to Scikit-Learn's GLM implementations on these datasets. These standard GLMs were trained on X^{train} and y^{train}.
- **Decoupled Regression.** We implemented decoupled regression in Scipy so that all components but the gradient calculation were similar. We calculated the gradients with the estimated ρ_i rather than the labeled training data X and y that would normally be used in a GLM. Thus, this model trained on X^{train} and the β_is.
- **Yoneoka et al 2015.** We are not aware of a publicly available software implementation of the methods in [135], so we implemented it ourselves using JAX to minimize the reported loss function for logistic regression meta-analysis. Solution formulas were available only for logistic regression, so we did not report performance results on the non-logistic tasks. This method used labeled training data. Thus, this model trained on X^{train}, y^{train}, and the β_is.
- **Naive (Univariate).** This method averages all reported regression coefficients, ignoring missing values (coefficients that were not studied in a particular paper). This method did not have access to labeled training data. Thus, this model trained on X^{train} and the β_is.

After training all models, we evaluated their loss on the training data (binary log-loss for logistic models, mean squared error for least-squares models, and Poisson deviance for Poisson models). We normalized all

losses so that 0 represented the loss of perfect predictions and 1 represented the loss of a constant mean prediction. We focused on training error because our goal was to ensure that decoupled regression achieved an equally good fit to the training data as standard GLMs; we deferred considerations of generalization performance to the eICU meta-analysis.

4.4.4 eICU Meta-Analysis

To evaluate a meta-analysis method for regression coefficients, we must perform a meta-analysis and test the performance of the resulting model on held-out data. One way to do this is to simulate the meta-analysis on a large dataset with a held-out test set. High-quality simulations would also involve learning different regression models in geographically or temporally distinct subsets of the data before combining them to represent the fact that meta-analysis often combines studies from many different times, geographic regions, or clinical settings [135]. Thus, our dataset of choice was the Philips eICU dataset [47], which combines data from 140,000 patient stays at over 200 intensive care units (ICUs) around the United States. ICU mortality is an important prediction target because it is (1) prevalent, with mortality rates in United States ICUs as high as 19 percent, (2) costly, with ICU expenditures constituting 14 percent of hospital costs annually, and (3) highly variable across patients and hospitals even after adjusting for baseline patient characteristics [26].

Each ICU in this dataset is annotated with a coarse geographic region and with a hospital identifier. Thus, we can introduce plausible domain shifts in "meta-analyzed" regression coefficients by region and hospital. This lets us precisely characterize the behavior of meta-analysis models under a range of realistic domain shifts we might expect to see in practice.

We split the dataset by geographic region into three sets: a training set, consisting of patient stays from the northeastern United States; a "study set," consisting of patient stays from the southern United States; and a test set, consisting of patients from the western United States. We used the training set as the unlabeled data in the decoupled regression framework, the study set to generate associations for the decoupled regression, and the test set to evaluate model performance. We generated univariate associations between each feature and the mortality outcome and tried to learn a mortality model using the unlabeled data and univariate associations alone.

We performed multiple meta-analyses and varied the number of patients per study to assess the sample size requirements of all methods. For each study i, we drew random study set hospitals one at a time and added each hospitals' patients to the sample for estimating odds-ratio i until our desired sample size was reached. This strategy should cause very high bias at the smallest sample sizes, as few if any studies will share patients; however, as the sample size grows to match the study set size, each univariate study will be performed on the exact same patients, reducing study-specific biases to zero.

We trained decoupled regression models both with and without regularization as well as an IPD model. We also displayed results from the naive associations model and a Scipy GLM trained on the *labeled* training set to give reference points for low- and high-quality predictions, respectively. The log-loss of each model on the test data was plotted as a function of the number of samples used in each study in Figure 4.2c.

4.4.5 Likelihood and AIC

Achieving good performance in our eICU meta-analysis was much easier with effective regularization; however, choosing the strength of this regularization required calculating log-likelihood and AIC for decoupled GLMs with a range of regularization parameters. AIC is difficult to calculate for arbitrary penalized GLMs but simple to calculate for L1 penalized models because degrees of freedom can be estimated as the number of non-zero parameters [137]. All that remains is to calculate the model's log-likelihood. This usually requires labeled data, but we assume access only to unlabeled data and the gradients of the log-likelihood.

Our access to log-likelihood gradients suggests a natural solution: we can use the gradient theorem to estimate the log-likelihood as the integral of its gradients. Specifically, the gradient theorem states that for a curve γ from $\mathbf{a}$ to $\mathbf{b}$, we have

$$\int_{\gamma} \nabla f(\mathbf{v}) \cdot d\mathbf{r} = f(\mathbf{b}) - f(\mathbf{a}).$$

Letting $\mathbf{b}$ be our estimated parameter vector $\hat{\beta}$ and f be the likelihood ℓ, we can rearrange to get

$$\ell(\hat{\beta}) = \int_{\gamma} \nabla \ell(\theta(\beta)) \cdot d\mathbf{r} + \ell(\beta_0) \tag{4.3}$$

for some starting parameter vector β_0 and a path γ interpolating β from β_0 to $\hat{\beta}$ (we simply use a straight-line path).

However, we still need a value of β_0 with known log-likelihood. The log-likelihood for a model that predicts any constant output (i.e., all coefficients are zero except for the bias) can be calculated assuming that the marginal distribution of the labels is known. Given this β_0, we can then estimate the log-likelihood $\ell(\hat{\beta})$ by using Equation 4.1 for the gradient calculation in Equation 4.3.

The final issue is that the formula in Equation 4.1 is proportional, and not exactly equal, to the gradients required. Nevertheless, we can simply calculate the constant of proportionality $\frac{1}{a(\phi)}$ by introducing another vector β_1 that makes a different constant prediction than β_0, calculating its corresponding log-likelihood, and then choosing the proportionality constant $\frac{1}{a(\phi)}$ so that Equation F gives

$$\ell(\beta_1) - \ell(\beta_0) = \frac{1}{a(\phi)} \int_{\gamma} (X^T y - X^T \hat{y}) \cdot d\mathbf{r}.$$

where γ is a straight line path interpolating β from β_0 to β_1.

The final likelihood is given by

$$\ell(\hat{\beta}) = \frac{1}{a(\phi)} \int_{\gamma} (X^T y - X^T \hat{y}) \cdot d\mathbf{r} + \ell(\beta_0) \tag{4.4}$$

where γ is a straight line path interpolating β from β_0 to $\hat{\beta}$.

Chapter 5

Conclusion

The field of machine learning in healthcare is full of promise. Clinical risk scores have the potential to make risk prediction a much easier task for physicians and to prevent huge amounts of illness, injury and death for patients. This work has presented several methods for addressing common data constraints in clinical prediction tasks, in the interest of broadening the range of clinical scenarios in which AI and machine learning are practical problem-solving tools. Across multiple clinical scenarios, including costly features at test time and limited data or absent labels at training time, we were able to substantially improve the performance and real-world usability of modeling methods for predicting patient risk.

Specific directions for future work on costly features include expanding approaches for calculating feature importance and performing feature selection. Much of the value of CoAI stems from the fact that it precomputes feature importance before selecting features, allowing new feature sets to be recomputed when costs or predictive budgets shift. However, summarizing importance in a single number for each feature inherently discards large amounts of information. What is the best way to precompute the value of features in a way that allows efficient downstream feature selection in a world of shifting costs and measurement budgets?

Future work on attribution priors is likely to involve many new specific functions to be used as priors; some such research has already been published! However, at a higher level, attribution priors is a cycle for communicating information about the model's behavior to a user and pushing the model to behave more in-line with the user's expectations. To this end, reliable methods for eliciting the priors of users and domain experts and mathematically encoding them as attribution priors are essential. We should not only focus on finding equations that improve model performance, we should strive to become better at helping humans articulate what beliefs about the world they would like to see encoded in their predictive models.

Finally, decoupled regression has presented a way to train generalized linear models that are consistent with published studies, but this framework should not be limited just to linear models. It is possible that other types of predictive modeling can be trained via sufficient statistics or maximum-likelihood to represent a good fit to published knowledge. It would also be valuable to study what types of statistics, when published in a given study, would best support the meta-analytic training of new, useful model types.

The idea of *real-world* clinical risk prediction has motivated much of this thesis; are the predictive methods we develop in computer science helpful to a clinician at the patient's bedside? Each of the three main projects discussed in this thesis is a step toward answering "yes". However, there are innumerable avenues for future research to improve the value of computer science to clinicians; these directions go far beyond the three broad categories discussed in this thesis. How to choose new directions? I found that one of the most rewarding parts of this thesis work was gathering data from emergency medical service providers about the degree to which data-gathering for risk prediction imposes a burden during the process of care. I believe any future work that attempts to improve patients' or providers' lives with AI must start with a good-faith effort to understand how predictive models affect their experiences in the health care system. The most interesting problems and best solutions in clinical risk prediction will be found through a truly interdisciplinary approach. I am happy to have spent four years immersed in interdisciplinary research collaborations at the University of Washington and in the broader machine learning for healthcare community, and I look forward to seeing the achievements of these communities in the years to come.

Bibliography

[1] MDCalc. *Frequently Asked Questions*. https://www.mdcalc.com/faq. Accessed: 2020-07-11. 2019.

[2] Alice Ferng. *Understanding and Creating Calculators for Medical Diagnoses: Exclusive Interview with MDCalc*. Medgadget. URL: https://www.medgadget.com/2018/06/understanding-and-creating-new-calculators-for-medical-diagnoses-exclusive-interview-with-mdcalc.html.

[3] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), p. 115.

[4] Varun Gulshan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs". In: *Jama* 316.22 (2016), pp. 2402–2410.

[5] Awni Y Hannun et al. "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network". In: *Nature medicine* 25.1 (2019), p. 65.

[6] Zachary C Lipton et al. "Learning to diagnose with LSTM recurrent neural networks". In: *International Conference on Machine Learning*. 2017.

[7] Scott M Lundberg et al. "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery". In: *Nature Biomedical Engineering* 2.10 (2018), p. 749.

[8] Andrew Ng. *Deep Learning for building AI systems*. Tutorial. 2016.

[9] Aya A Mitani and Sebastien Haneuse. "Small data challenges of studying rare diseases". In: *JAMA network open* 3.3 (2020), e201965–e201965.

[10] Stéphanie Nguengang Wakap et al. "Estimating cumulative point prevalence of rare diseases: analysis of the Orphanet database". In: *European Journal of Human Genetics* 28.2 (2020), pp. 165–173.

[11] Ethan Weinberger, Joseph Janizek, and Su-In Lee. "Learning Deep Attribution Priors Based On Prior Knowledge". In: *arXiv preprint arXiv:1912.10065* (2019).

[12] Alex Tseng, Avanti Shrikumar, and Anshul Kundaje. "Fourier-transform-based attribution priors improve the interpretability and stability of deep learning models for genomics". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1913–1923. URL: https://proceedings.neurips.cc/paper/2020/file/1487987e862c44b91a0296cf3866387e-Paper.pdf.

[13] Gabriel Erion et al. "CoAI: Cost-Aware Artificial Intelligence for Health Care". In: *medRxiv* (2021). DOI: 10.1101/2021.01.19.21249356. eprint: https://www.medrxiv.org/content/early/2021/01/20/2021.01.19.21249356.full.pdf. URL: https://www.medrxiv.org/content/early/2021/01/20/2021.01.19.21249356.

[14] Washington State Department of Health. "Trauma in Washington State: A chart report of the first 15 years, 1995–2009". In: (2011).

[15] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[16] Sven Peter et al. "Cost efficient gradient boosting". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1551–1561.

[17] Jaromır Janisch, Tomáš Pevn, and Viliam Lis. "Classification with costly features using deep reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3959–3966.

[18] Jaromır Janisch, Tomáš Pevn, and Viliam Lis. "Classification with costly features as a sequential decision-making problem". In: *Machine Learning* (2020), pp. 1–29.

[19] D Frith et al. "Definition and drivers of acute traumatic coagulopathy: clinical and experimental investigations". In: *Journal of Thrombosis and Haemostasis* 8.9 (2010), pp. 1919–1925.

[20] Biswadev Mitra et al. "Acute coagulopathy and early deaths post major trauma". In: *Injury* 43.1 (2012), pp. 22–25.

[21] Karim Brohi, Mitchell J Cohen, and Ross A Davenport. "Acute coagulopathy of trauma: mechanism, identification and effect". In: *Current opinion in critical care* 13.6 (2007), pp. 680–685.

[22] Satoshi Gando and Mineji Hayakawa. "Pathophysiology of trauma-induced coagulopathy and management of critical bleeding requiring massive transfusion". In: *Seminars in thrombosis and hemostasis*. Vol. 42. 02. Thieme Medical Publishers. 2016, pp. 155–165.

[23] Ross Davenport et al. "Functional definition and characterisation of acute traumatic coagulopathy". In: *Critical care medicine* 39.12 (2011), p. 2652.

[24] Ithan D Peltan et al. "Development and validation of a prehospital prediction model for acute traumatic coagulopathy". In: *Critical Care* 20.1 (2016), p. 371.

[25] Biswadev Mitra et al. "Early prediction of acute traumatic coagulopathy". In: *Resuscitation* 82.9 (2011), pp. 1208–1213.

[26] Neil Halpern. *Critical Care Statistics*. 2019. URL: `https://www.sccm.org/Communications/Critical-Care-Statistics`.

[27] Alistair EW Johnson, Andrew A Kramer, and Gari D Clifford. "A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy". In: *Critical care medicine* 41.7 (2013), pp. 1711–1718.

[28] Christopher W Seymour et al. "Assessment of clinical criteria for sepsis: for the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". In: *Jama* 315.8 (2016), pp. 762–774.

[29] Henry W Miller. "Plan and operation of the health and nutrition examination survey, United States, 1971-1973". In: *DHEW publication no.(PHS)-Dept. of Health, Education, and Welfare (USA)* (1973).

[30] Nicholas A Christakis and Theodore J Iwashyna. "Attitude and self-reported practice regarding prognostication in a national sample of internists". In: *Archives of Internal Medicine* 158.21 (1998), pp. 2389–2395.

[31] P. Rui and T. Okeyode. "National Ambulatory Medical Care Survey: 2016 National Summary Tables". In: (2016).

[32] Sei J Lee et al. "Development and validation of a prognostic index for 4-year mortality in older adults". In: *Jama* 295.7 (2006), pp. 801–808.

[33] Roland M du Bois et al. "Ascertainment of individual risk of mortality for patients with idiopathic pulmonary fibrosis". In: *American journal of respiratory and critical care medicine* 184.4 (2011), pp. 459–466.

[34] Bartolome R Celli et al. "The body-mass index, airflow obstruction, dyspnea, and exercise capacity index in chronic obstructive pulmonary disease". In: *New England Journal of Medicine* 350.10 (2004), pp. 1005–1012.

[35] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[36] Laurent Perron and Vincent Furnon. *OR-Tools (version 7.2)*. URL: `https://developers.google.com/optimization/`.

[37] Ian Covert and Su-In Lee. "Improving KernelSHAP: Practical Shapley Value Estimation via Linear Regression". In: *arXiv preprint arXiv:2012.01536* (2020).

[38] Ian Covert, Scott Lundberg, and Su-In Lee. "Understanding global feature contributions through additive importance measures". In: *arXiv preprint arXiv:2004.00668* (2020).

[39] Scott M. Lundberg et al. *Explainable AI for Trees: From Local Explanations to Global Understanding*. 2019. arXiv: `1905.04610 [cs.LG]`.

[40] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4765–4774.

[41] Hugh Chen, Scott Lundberg, and Su-In Lee. "Explaining Models by Propagating Shapley Values of Local Components". In: *arXiv preprint arXiv:1911.11888* (2019).

[42] Maytal Saar-Tsechansky and Foster Provost. "Handling missing values when applying classification models". In: *Journal of machine learning research* 8.Jul (2007), pp. 1623–1657.

[43] Kaiyuan Li et al. "A Machine Learning–Based Model to Predict Acute Traumatic Coagulopathy in Trauma Patients Upon Emergency Hospitalization". In: *Clinical and Applied Thrombosis/Hemostasis* 26 (2020), p. 1076029619897827.

[44] Timothy C Nunez et al. "Early prediction of massive transfusion in trauma: simple as ABC (assessment of blood consumption)?" In: *Journal of Trauma and Acute Care Surgery* 66.2 (2009), pp. 346–352.

[45] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software, Articles* 45.3 (2011), pp. 1–67. ISSN: 1548-7660. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/v045/i03.

[46] Abigail R Wheeler et al. "Development of prehospital assessment findings associated with massive transfusion". In: *Transfusion* (2020).

[47] Tom J Pollard et al. "The eICU Collaborative Research Database, a freely available multi-center database for critical care research". In: *Scientific data* 5 (2018).

[48] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3146–3154.

[49] Centers for Medicare and Medicaid Services. *Clinical Laboratory Fee Schedule Files - CY 2019 Q3 Release*. URL: https://cms.gov/Medicare/Medicare-Fee-for-Service-Payment/ClinicalLabFeeSched/Clinical-Laboratory-Fee-Schedule-Files.html.

[50] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. Journal of Machine Learning Research. 2017, pp. 3319–3328.

[51] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[52] John Ashworth Nelder and Robert WM Wedderburn. "Generalized linear models". In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pp. 370–384.

[53] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[54] François Chollet et al. *Keras*. https://keras.io. 2015.

[55] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[56] Kirstin Early, Stephen E Fienberg, and Jennifer Mankoff. "Test time feature ordering with FOCUS: interactive predictions with minimal user burden". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 992–1003.

[57] Feng Nan, Joseph Wang, and Venkatesh Saligrama. "Pruning random forests for prediction on a budget". In: *Advances in neural information processing systems*. 2016, pp. 2334–2342.

[58] Feng Nan and Venkatesh Saligrama. "Adaptive classification for prediction under a budget". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4727–4737.

[59] Yu-Shao Peng et al. "Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7322–7331.

[60] Mohammad Kachuee et al. "Opportunistic Learning: Budgeted Cost-Sensitive Learning from Data Streams". In: *International Conference on Learning Representations*. 2019.

[61] Mohammad Kachuee et al. "Nutrition and Health Data for Cost-Sensitive Learning". In: *arXiv preprint arXiv:1902.07102* (2019).

[62] Gabriel G. Erion et al. "Improving performance of deep learning models with axiomatic attribution priors and expected gradients". In: *Nature machine intelligence* (2021).

[63] Erik Štrumbelj and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and information systems* 41.3 (2014), pp. 647–665.

[64] 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*. IEEE. 2016, pp. 598–617.

[65] Scott M Lundberg et al. "From local explanations to global understanding with explainable AI for trees". In: *Nature: Machine Intelligence* (2020).

[66] Rory Sayres et al. "Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy". In: *Ophthalmology* 126.4 (2019), pp. 552–564.

[67] John R Zech et al. "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study". In: *PLoS medicine* 15.11 (2018), e1002683.

[68] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. "Right for the right reasons: Training differentiable models by constraining their explanations". In: *arXiv preprint arXiv:1703.03717* (2017).

[69] Patrick Schramowski et al. "Making deep neural networks right for the right scientific reasons by interacting with their explanations". In: *Nature Machine Intelligence* 2.8 (2020), pp. 476–486.

[70] Andrew Ilyas et al. "Adversarial Examples Are Not Bugs, They Are Features". In: *arXiv preprint arXiv:1905.02175* (2019).

[71] Frederick Liu and Besim Avci. "Incorporating Priors with Feature Attribution on Text Classification". In: *arXiv preprint arXiv:1906.08286* (2019).

[72] Jiefeng Chen et al. "Robust Attribution Regularization". In: *arXiv preprint arXiv:1905.09957* (2019).

[73] Laura Rieger et al. "Interpretations are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge". In: *Proceedings of the 37 th International Conference on Machine Learning*. 2020.

[74] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* 2 (2010), p. 18.

[75] Fuxun Yu et al. "Towards Robust Training of Neural Networks by Regularizing Adversarial Gradients". In: *arXiv preprint arXiv:1805.09370* (2018).

[76] Daniel Jakubovitz and Raja Giryes. "Improving DNN robustness to adversarial attacks using Jacobian regularization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 514–529.

[77] Kevin Roth et al. "Adversarially robust training through structured gradient regularization". In: *arXiv preprint arXiv:1805.08736* (2018).

[78] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 618–626.

[79] 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.

[80] Daniel Smilkov et al. "Smoothgrad: removing noise by adding noise". In: *arXiv preprint arXiv:1706.03825* (2017).

[81] Ruth C Fong and Andrea Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3429–3437.

[82] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[83] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[84] Benjamin Recht et al. "Do ImageNet Classifiers Generalize to ImageNet?" In: *arXiv preprint arXiv:1902.10811* (2019).

[85] Dimitris Tsipras et al. "Robustness may be at odds with accuracy". In: *arXiv preprint arXiv:1805.12152* (2018).

[86] Hongyang Zhang et al. "Theoretically principled trade-off between robustness and accuracy". In: *arXiv preprint arXiv:1901.08573* (2019).

[87] Wei Cheng et al. "Graph-regularized dual Lasso for robust eQTL mapping". In: *Bioinformatics* 30.12 (June 2014), pp. i139–i148. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu293. eprint: http://oup.prod.sis.lan/bioinformatics/article-pdf/30/12/i139/17347402/btu293.pdf. URL: https://doi.org/10.1093/bioinformatics/btu293.

[88] Jeffrey W Tyner et al. "Functional genomic landscape of acute myeloid leukaemia". In: *Nature* 562.7728 (2018), p. 526.

[89] Casey S Greene et al. "Understanding multicellular function and disease with human tissue-specific networks". In: *Nature genetics* 47.6 (2015), p. 569.

[90] Thomas N Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *CoRR* abs/1609.0 (2016). arXiv: 1609.02907. URL: http://arxiv.org/abs/1609.02907.

[91] Aravind Subramanian et al. "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43 (Oct. 2005), 15545 LP –15550. DOI: 10.1073/pnas.0506580102. URL: http://www.pnas.org/content/102/43/15545.abstract.

[92] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300. ISSN: 00359246. URL: http://www.jstor.org/stable/2346101.

[93] Jiangying Liu et al. "Meis1 is critical to the maintenance of human acute myeloid leukemia cells independent of MLL rearrangements". In: *Annals of Hematology* 96.4 (Apr. 2017), pp. 567–574. ISSN: 1432-0584. DOI: 10.1007/s00277-016-2913-6. URL: https://doi.org/10.1007/s00277-016-2913-6.

[94] Peter J M Valk et al. "Prognostically Useful Gene-Expression Profiles in Acute Myeloid Leukemia". In: *New England Journal of Medicine* 350.16 (2004), pp. 1617–1628. DOI: 10.1056/NEJMoa040465. URL: https://doi.org/10.1056/NEJMoa040465.

[95] Jean Feng and Noah Simon. "Sparse-input neural networks for high-dimensional nonparametric regression and classification". In: *arXiv preprint arXiv:1711.07592* (2017).

[96] Simone Scardapane et al. "Group sparse regularization for deep neural networks". In: *Neurocomputing* 241 (2017), pp. 81–89.

[97] 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.

[98] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. Journal of Machine Learning Research. 2017, pp. 3145–3153.

[99] Niall Hurley and Scott Rickard. "Comparing measures of sparsity". In: *IEEE Transactions on Information Theory* 55.10 (2009), pp. 4723–4741.

[100] Dornoosh Zonoobi, Ashraf A Kassim, and Yedatore V Venkatesh. "Gini index as sparsity measure for signal reconstruction from compressive samples". In: *IEEE Journal of Selected Topics in Signal Processing* 5.5 (2011), pp. 927–932.

[101] Henry W Miller. "Plan and operation of the health and nutrition examination survey, United States, 1971-1973". In: *DHEW publication no.(PHS)-Dept. of Health, Education, and Welfare (USA)* (1973).

[102] Alexander Binder et al. "Layer-wise relevance propagation for neural networks with local renormalization layers". In: *International Conference on Artificial Neural Networks*. Springer. 2016, pp. 63–71.

[103] Eric J Friedman. "Paths and consistency in additive cost sharing". In: *International Journal of Game Theory* 32.4 (2004), pp. 501–518.

[104] Hongyi Zhang et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[105] Johnathan M Bardsley. "Laplace-distributed increments, the Laplace prior, and edge-preserving regularization". In: *J. Inverse Ill-Posed Probl* (2012).

[106] Martín Abadi et al. "Tensorflow: A system for large-scale machine learning". In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, pp. 265–283.

[107] Yifei Lou et al. "A weighted difference of anisotropic and isotropic total variation model for image processing". In: *SIAM Journal on Imaging Sciences* 8.3 (2015), pp. 1798–1823.

[108] Yuying Shi and Qianshun Chang. "Efficient algorithm for isotropic and anisotropic total variation deblurring and denoising". In: *Journal of Applied Mathematics* 2013 (2013).

[109] 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)*. IEEE. 2015, pp. 730–734.

[110] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[111] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[112] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.

[113] Kristina Preuer et al. "DeepSynergy: predicting anti-cancer drug synergy with Deep Learning". In: *Bioinformatics* 34.9 (2018), pp. 1538–1546. DOI: 10.1093/bioinformatics/btx806. URL: http://dx.doi.org/10.1093/bioinformatics/btx806.

[114] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 00359246. URL: http://www.jstor.org/stable/2346178.

[115] Pascal Sturmfels, Gabriel Erion, and Joseph D. Janizek. *suinleelab/attributionpriors: Nature Machine Intelligence code*. Version v1.0.0. Mar. 2021. DOI: 10.5281/zenodo.4608599. URL: https://doi.org/10.5281/zenodo.4608599.

[116] Harris Drucker and Yann Le Cun. "Improving generalization performance using double backpropagation". In: *IEEE Transactions on Neural Networks* 3.6 (1992), pp. 991–997.

[117] Shayak Sen et al. "Supervising Feature Influence". In: *arXiv preprint arXiv:1803.10815* (2018).

[118] W James Murdoch, Peter J Liu, and Bin Yu. "Beyond word importance: Contextual decomposition to extract interactions from LSTMs". In: *arXiv preprint arXiv:1801.05453* (2018).

[119] Laura Rieger and Chandan Singh. *CDEP Github*. https://github.com/laura-rieger/deep-explanation-penalization. Accessed: 2020-07-17. 2020.

[120] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[121] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[122] Nathan Silberman and Sergio Guadarrama. "Tensorflow-slim image classification model library". In: (2016).

[123] Ali Mortazavi et al. "Mapping and quantifying mammalian transcriptomes by RNA-Seq". In: *Nature Methods* 5 (May 2008), p. 621. URL: https://doi.org/10.1038/nmeth.1226%20http://10.0.4.14/nmeth.1226%20https://www.nature.com/articles/nmeth.1226%7B%5C#%7Dsupplementary-information.

[124] Jeffrey T Leek and John D Storey. "Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis". In: *PLOS Genetics* 3.9 (Sept. 2007), pp. 1–12. DOI: 10.1371/journal.pgen.0030161. URL: https://doi.org/10.1371/journal.pgen.0030161.

[125] Eric Steinberg et al. *Critical Review: COVID-19 Calculators during Extreme Resource-Limited Situations*. 2020. URL: https://www.mdcalc.com/covid-19/calculators-extreme-resource-limited-situations.

[126] Christopher M. Petrilli et al. "Factors associated with hospitalization and critical illness among 4,103 patients with COVID-19 disease in New York City". In: *medRxiv* (2020). DOI: 10.1101/2020.04.08.20057794. eprint: https://www.medrxiv.org/content/early/2020/04/11/2020.04.08.20057794.full.pdf. URL: https://www.medrxiv.org/content/early/2020/04/11/2020.04.08.20057794.

[127] Akhil Vaid et al. "Machine Learning to Predict Mortality and Critical Events in COVID-19 Positive New York City Patients". In: *medRxiv* (2020). DOI: 10.1101/2020.04.26.20073411. eprint: https://www.medrxiv.org/content/early/2020/04/28/2020.04.26.20073411.full.pdf. URL: https://www.medrxiv.org/content/early/2020/04/28/2020.04.26.20073411.

[128] Wenhua Liang et al. "Development and validation of a clinical risk score to predict the occurrence of critical illness in hospitalized patients with COVID-19". In: *JAMA internal medicine* 180.8 (2020), pp. 1081–1089.

[129] Betsy Jane Becker, Meng-Jia Wu, et al. "The synthesis of regression slopes in meta-analysis". In: *Statistical science* 22.3 (2007), pp. 414–429.

[130] Thomas PA Debray et al. "Aggregating published prediction models with individual participant data: a comparison of different approaches". In: *Statistics in medicine* 31.23 (2012), pp. 2697–2712.

[131] Meng-Jia Wu and Betsy Jane Becker. "Synthesizing regression results: a factored likelihood method". In: *Research synthesis methods* 4.2 (2013), pp. 127–143.

[132] RD Riley et al. "Multivariate meta-analysis using individual participant data". In: *Research synthesis methods* 6.2 (2015), pp. 157–174.

[133] Matthieu Resche-Rigon et al. "Multiple imputation for handling systematically missing confounders in meta-analysis of individual participant data". In: *Statistics in medicine* 32.28 (2013), pp. 4890–4905.

[134] Daisuke Yoneoka and Masayuki Henmi. "Synthesis of linear regression coefficients by recovering the within-study covariance matrix from summary statistics". In: *Research synthesis methods* 8.2 (2017), pp. 212–219.

[135] Daisuke Yoneoka et al. "Synthesis of clinical prediction models under different sets of covariates with one individual patient data". In: *BMC medical research methodology* 15.1 (2015), pp. 1–11.

[136] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.

[137] Hui Zou, Trevor Hastie, Robert Tibshirani, et al. "On the "degrees of freedom" of the lasso". In: *The Annals of Statistics* 35.5 (2007), pp. 2173–2192.