Regional Tree Regularization for Interpretability in Black Box Models

Anonymous

ABSTRACT

The lack of interpretability remains a barrier to the adoption of deep models. Recently, tree-regularization has been proposed as a method to encourage deep neural networks to resemble simple decision trees without significant compromises in accuracy. However, it is often unreasonable to imagine that one could describe a sufficiently complex problem with a reasonably-sized tree. In this work, we explicitly regularize deep models to be closely approximated by decision trees within (intuitive) regions. Across many data sets, including two real health applications, we demonstrate that our approach yields models that admit simpler explanations without sacrificing predictive power, enabling understanding by experts.

CCS CONCEPTS

• **Computing methodologies → Regularization**; *Classification and regression trees*; *Neural networks.*

KEYWORDS

Tree Regularization, Interpretability, Deep Neural Networks

ACM Reference Format:

Anonymous. 1997. Regional Tree Regularization for Interpretability in Black Box Models. In *Proceedings of ACM SIGKDD conference (KDD'19),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Deep models have become the state-of-the-art in applications ranging from image classification [15] to game playing [20], and they are poised to advance prediction in real-world domains such as medicine and healthcare [7, 8, 19]. However, understanding when a model's outputs can be trusted and how the model might be improved remains a challenge. In particular, Chen et al. [3] discuss how these challenges inhibit the adoption of deep models in the clinical community. Interpretability is essential for incorporating domain knowledge and enabling humans to audit predictions.

As such, many efforts have been devoted to extracting explanation from deep models post-hoc, often through gradients, perturbation analysis, or distillation [2, 25, 27]. However, for an arbitrary deep model, these explanations might be too complex for a human to understand. Thus, recent work has instead optimized deep models directly for interpretability [26, 29]. In particular, Wu et al. [29] train a deep model to be closely-approximated by a decision tree, which can then serve as a *human-simulatable* explanation [16]. Simulatable explanations allows humans to, in reasonable time, combine inputs and explanation to produce outputs and thus form a foundation for auditing and correcting predictions.

An important factor for any explanation is *when* it is valid. Previous work has focused on two opposing regimes. Works on *global* explanation (e.g. Wu et al. [29]) return a single explanation for the *entire* model. Unfortunately, if the explanation is simple enough to be simulatable, then it is unlikely to be faithful to the deep model across all inputs. In contrast, works on *local* explanation (e.g. Ribeiro et al. [25], Selvaraju et al. [27]) seek to explain individual predictions. These explanations lack a sense of generality, as individual glimpses to the model's behavior can fail to capture a larger pattern. More troubling, all local approaches do consider a (perhaps infinitesimal) region around an input—but only implicitly. Thus a user, having seen an explanation for an input x, cannot easily know whether the same logic should hold for a nearby point x'. This ambiguity can lead to mistaken assumptions.

In this work, we consider a middle-ground: *regional* explanations that constrain the model independently across a set of predefined regions of the input space. This form of explanation is consistent with those of humans, whose models are typically context-dependent [18]. For example, physicians in the intensive care unit do not expect treatment rules to be the same across different categories of patients. Constraining each region to be human-simulatable allows the deep model to be more complex than a global constraint, while still allowing the it to capture patterns across all the data. However, optimization with many regions is challenging.

In the following, we describe a computationally tractable and reliable approach to optimizing models for simple regional explanations. Specifically, we (1) describe how to jointly train a deep model that has both high accuracy and supports simple, faithful explanations across user-specified regions, (2) introduce inference innovations for stability in optimization, and (3) demonstrate that that our approach achieves comparable performance to more complex models while learning a much simpler decision function.

2 RELATED WORK

Global Interpretability Given an *already-trained* black box model, there are many approaches to extract what the model has learned. Works such as Mordvintsev et al. [21] expose the features a representation encodes, but do not expose the logic. Amir and Amir [1], Kim et al. [10] provide an informative set of examples that summarize the system. Other directions [29] attempt to regularize a complex model to behave like a simpler family of models. For example, model distillation compress a source network into a smaller target neural network [6]. However, the distilled model may still not be interpretable, and even when it is, a small model is unlikely to faithfully describe a more complex model.

Local Interpretability In contrast, these approaches focus on providing explanation for a specific input. Ribeiro et al. [25] show that

KDD'19, August 2019, Anchorage, Alaska, USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

using the weights of a sparse linear model, one can explain the decisions of a black box model in a small area near a fixed data point. Similarly, instead of a linear model, Singh et al. [28] and Koh and Liang [12] output a simple program or an influence function, respectively. Other approaches still have used input gradients (which can be thought of as infinitesimal perturbations) to characterize the local space [17, 27]. However, the notion of a local region in these works is both very small (sometimes infinitesimal) and often implicit; it does not match with human notions of contexts [18].

Optimizing for Interpretability Deep models have many local optima, some of which may admit more human-simulatable explanations than others. Instead of interpreting a model post-hoc, an alternative is to optimize a measure of interpretability alongside predictive performance. Ross et al. [26], Wu et al. [29] pose two paths forward: include input gradient explanations or decision tree explanations in the objective function. As a result, models are encouraged to find "more interpretable" minima. Similarly, Krening et al. [14] jointly train a model to provide a verbal explanation alongside an image classifier. In this paper, we push these ideas forward by optimizing for "regional" interpretability.

3 BACKGROUND AND NOTATION

We consider supervised learning tasks given a dataset of N labeled examples, $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$, where each example (indexed by n) has an input feature vector $\mathbf{x}_n = \{x_{n,1}, x_{n,2}, ..., x_{n,P}\} \in \mathcal{X}^P$ and a target output vector $\mathbf{y}_n = \{y_{n,1}, y_{n,2}, ..., y_{n,Q}\} \in \mathcal{Y}^Q$ where $P, Q \in \mathbb{N}$ are the dimensionalities. We assume each target $y_{n,q} \in \mathbf{y}_n$ is a discrete indicator selecting one of a fixed number of possible classes, though it is simple to extend to continuous domains.

Multi-Layer Perceptrons We consider multi-layer perceptrons (MLPs) as the representative of a deep model; that being said, the core ideas presented here can be easily extended to other architectures including recurrent and convolutional networks. Formally, a perceptron defines a parameterized function, $\hat{\mathbf{y}}_n = f(\mathbf{x}_n; \theta), f : \mathcal{X}^P \to \mathcal{Y}^Q$ where $\hat{\mathbf{y}}_n = \{\hat{y}_{n,1}, \hat{y}_{n,2}, ..., \hat{y}_{n,Q}\}$ are estimates for the true target $\mathbf{y}_n$. The vector $\theta \in \Theta$ represents all parameters of the neural network. Given a dataset $\mathcal{D}$, the goal is to recover the best parameters θ to minimize the objective,

$$\arg\min_{\theta \in \Theta} \sum_{n=1}^{N} \mathcal{L}(\mathbf{y}_n, f(\mathbf{x}_n; \theta)) + \lambda \Omega(\theta). \tag{1}$$

For binary targets $\mathbf{y}_n$, the logistic loss is an effective choice for $\mathcal{L}(\cdot)$. The function $\Omega : \Theta \to \mathbb{R}$ represents a regularization penalty, with scalar strength $\lambda \in \mathbb{R}^+$. Standard penalty functions include the L1 or L2 norm of θ. In the following, we shall refer to this predictor $f(\cdot; \theta)$ as our *target neural model*.

Global Tree Regularization Wu et al. [29] introduced a regularization term that penalizes models for being hard to simulate where (human-)simulatability is measured by the "size" (or complexity) of the decision tree that best approximates the target neural model. The tree complexity is defined to be the *average decision path length* (abbreviated as APL), or the average number of decision nodes that must be touched to make a prediction for an example $\mathbf{x}_n \in \mathcal{D}$. Formally, the penalty term is written as:

$$\Omega^{\texttt{global}}(\theta) \triangleq \texttt{PathLength}(\{\mathbf{x}_n\}_{n=1}^{N}, f(\cdot), h) \tag{2}$$

where N is the size of $\mathcal{D}$, f is the target neural model, h is a decision tree hyperparameter, and PathLength is defined in Alg. 1. Specifially, from Alg. 1, TrainTree refers to any algorithm to fit a decision tree given input and output pairs (e.g. CART or ID3). PruneTree refers to removing "unnecessary" subtrees that do not effect prediction. Note that a disjoint portion of the dataset is reserved for pruning. GetDepth is a subroutine that returns the depth of the leaf node associated with an input example $\mathbf{x}_n$; in other words, it is the length of the trajectory from root to leaf.

Algorithm 1 PathLength

```
Require:
    $f(\cdot, \theta)$: discrete prediction function, with parameters $\theta$
    $\{\mathbf{x}_i\}_{i=1}^{N}$: a set of $N$ input examples
    $N_{\text{train}}$: number of training examples
    $h$: minimum number of samples to define a leaf node
function PathLength($\{\mathbf{x}_i\}_{i=1}^{N}, f, h$)
    $\hat{\mathbf{y}}_i = f(\mathbf{x}_i, \theta)$
    $T = $ TrainTree($\{\mathbf{x}_i, \hat{\mathbf{y}}_i\}_{i=1}^{N_{\text{train}}}, h$)
    $T = $ PruneTree($T, \{\mathbf{x}_i, \hat{\mathbf{y}}_i\}_{i=N_{\text{train}}}^{N}$)
    $a = 0$
    for $i$ in 1:$N$ do
        $a = a + $ GetDepth($T, \mathbf{x}_i$)
    return $a/N$
```

However, TrainTree is not differentiable, so Eqn. 2 cannot be in the optimization objective. Instead, Wu et al. [29] introduce a surrogate regularizer $\hat{\Omega}^{\texttt{global}} : \Theta \to \mathbb{R}^+$, that maps a parameter vector from a target neural model to an estimate of the APL. In practice, $\hat{\Omega}^{\texttt{global}}(\cdot)$ is a small neural network. Wu et al. [29] refer to this as the *surrogate model*. Intuitively, the problem of optimizing trees has become a supervised problem.

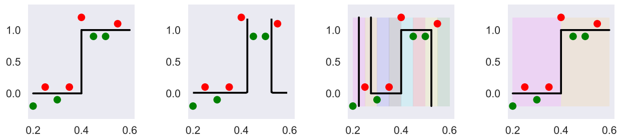
To train the surrogate model, they collect a dataset of parameters $\mathcal{D}_\theta = \{\theta_j, \Omega^{\texttt{global}}(\theta_j)\}_{j=1}^{J}$ from every gradient step in training the target neural model. They then optimize the following objective:

$$\arg\min_{\phi \in \Phi} \sum_{j=1}^{J} (\Omega^{\texttt{global}}(\theta_j) - \hat{\Omega}^{\texttt{global}}(\theta_j; \phi))^2 \tag{3}$$

Critically $\phi \in \Phi$, the parameters of the surrogate model, are a function of θ. Every few gradient steps in training the target neural model, we freeze θ and optimize ϕ to completion. This represents updating the mapping, $\hat{\Omega}^{\texttt{global}}$, as the target neural model changes.

4 REGIONAL FAITHFUL EXPLANATIONS

Attempts at global summaries have to trade between being human-simulatable and being faithful to the underlying model. In this section, we introduce a more fine-grain definition of interpretability. We assume that the space of inputs $\mathcal{X}^P$ can be divided into a set of R regions $X_1, ..., X_R$ where $\cup_{r=1}^{R} X_r \subseteq \mathcal{X}$ which may have different rules. For example, an intensivist may already cognitively consider patients in the sick intensive care unit (ICU) as belonging to a different category than patients in the cardiac ICU. Analogously, biologists may be happy with different models for classifying diseases in deciduous and coniferous plants. In this work, we assume



(a) True **(b) Global** **(c) Local** **(d) Regional**

Figure 1: We show the differences between global (b), local (c), and regional (d) tree regularization using a synthetic classification task. (a) shows the true decision boundary. Red and green points represent the training dataset. Lightly colored areas represent regions. In (b), the model is overregularized and ignores underlying structure. In (c), regions are made as small as possible to simulate locality—resulting in highly variable rules for nearby points. Regional tree regularization (d) provides an interpretable middle ground.

that the regions are defined either by (1) domain knowledge or (2) statistical approaches such as clustering.

We seek *regionally-faithful regularization*, meaning that the target neural model is "simple" in each region. This notion of explanations for different parts of the input space is consistent with cognitive science literature that suggests that humans build local (not global) models of the world [18]. Thus, it is sufficient for deep models to be locally simple for humans to understand it, where locality corresponds to regions defined by human contexts. We emphasize that regional and local explanations are conceptually different: the latter concerns itself with behavior within an ϵ-ball around a single data point, $\mathbf{x}_n \in \mathcal{X}$. It makes no claims about general behavior across data points. In contrast, regional explanations are faithful to a larger subset of $\mathcal{X}$, enabling stronger generalization.

As a preview of our model, Fig. 1 shows an example to highlight the distinction between global, local, and regional tree regularization: global explanations (b) possess no information about the input space and have to choose from a large set of possible solutions; local explanations (c) produce simple decision boundaries around each data point but fail to capture global relationships; regional explanations (d) share the benefits of (b) and (c). In this example, (d) captures the true decision boundary (a).

4.1 Regional Tree Regularization

We now introduce regional tree regularization, a penalty term for deep models that approximates its behavior with a decision tree per region. We choose decision trees as our form of explanation as they are capable of modeling nonlinearity while remaining human-simulatable. In particular, we require that each decision tree be *faithful* to the region; that is, it must match the underlying model some proportion (close to 1) of the time. The question, then, is how we can adjust the underlying model such that a set of faithful decision trees is relatively simple?

Formally, we denote a function $\Omega^{\texttt{regional}} : \Theta \rightarrow \mathbb{R}$ defined as,

$$\Omega^{\texttt{regional}}(\theta) \triangleq \frac{1}{R}\sum_{r=1}^{R} w_r \cdot \Omega_r^{\texttt{regional}}(\theta) \tag{4}$$

where $\Omega_r^{\texttt{regional}} : \Theta \rightarrow \mathbb{R}$ can be written as,

$$\Omega_r^{\texttt{regional}}(\theta) \triangleq \texttt{PathLength}(X_r, f(\cdot), h_r). \tag{5}$$

R is number of regions that partition $\mathcal{X}$, h_r is a hyperparameter for the decision tree in region r, and each $w_r \in \mathbb{R}$ is a weight attached to region r. We use these weights to balance regions of different sizes as we do not wish to equally penalize smaller regions with larger ones. In particular, $\Omega_r^{\texttt{regional}}$ is equivalent to $\Omega^{\texttt{global}}$ on the subset X_r as opposed to the full dataset $\mathcal{D}$.

Referring to Alg. 1 for computing APL, we define TrainTree as the `DecisionTreeClassifier` module distributed in Python's Scikit-Learn with a fixed random seed. Notably, TrainTree takes a single hyperparameter, h, that defines the minimum number of elements required to define a leaf node in the decision tree. Choosing h to be too small leads to overfitting, producing un-simulatable trees. Choosing h to be too big results in trivial trees, hindering optimization. We use a heuristic,

$$h_r = \min\{\frac{N_r}{C}, 1\} \tag{6}$$

where N_r is the size of X_r (the training set in region r) and C is a constant (dependent on the dataset). We do not use the same h_r in training and testing as the test set is often much smaller than the training set, so h_r should adjust accordingly (C remains the same).

4.2 Differentiable Decision-Tree Loss

Unfortunately, the objective in Eqn. 4 is not differentiable with respect to θ, as derivatives cannot flow through CART. As such, gradient descent is not immediately applicable. Like Wu et al. [29], we introduce a *surrogate* loss function $\hat{\Omega}_r^{\texttt{regional}} : \Theta \rightarrow \mathbb{R}^+$ that maps a parameter vector $\theta \in \Theta$ to an *estimate* of $\Omega_r^{\texttt{regional}}(\cdot)$, the APL in region r. In our case, $\hat{\Omega}_r^{\texttt{regional}}(\cdot)$ is implemented as a shallow multi-layer perceptron, and is thus differentiable:

$$\hat{\Omega}_r^{\texttt{regional}}(\theta; \phi_r) \triangleq \texttt{MLP}_r(\theta; \phi_r) \tag{7}$$

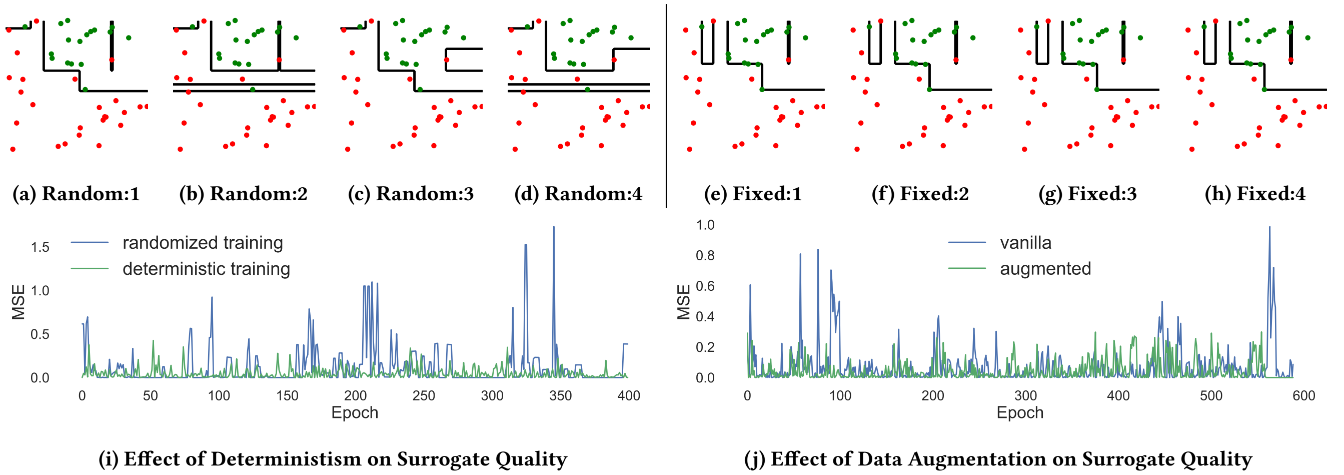
where $\phi_r \in \Phi$ are the parameters of the surrogate network for region r. This implies that there are R surrogate networks, each one estimating the average path length in a single region. In other words, we approximate Eqn. 4 as follows,

$$\hat{\Omega}_{\texttt{local}}(\theta; \phi_1, ..., \phi_R) \triangleq \frac{1}{R}\sum_{r=1}^{R} \texttt{MLP}_r(\theta; \phi_r) \tag{8}$$

The R surrogate models can be trained jointly. Specifically, we fit $\hat{\Omega}_r^{\texttt{regional}}(\theta)$ by minimizing a mean squared error loss,

$$\arg\min_{\phi_r} \sum_{j=1}^{J} (\Omega_r^{\texttt{regional}}(\theta_j) - \hat{\Omega}_r^{\texttt{regional}}(\theta_j, \phi_r))^2 \tag{9}$$

for all $r = 1, ..., R$ where θ_j is sampled from a dataset of J known parameter vectors and their true APLs: $\mathcal{D}_r^{\theta} = \{\theta_j, \Omega_r^{\texttt{regional}}(\theta_j)\}_{j=1}^{J}$. This dataset can be assembled using the candidate θ vectors obtained every gradient step while training our target neural model $f(\cdot, \theta)$. For R regions, we curate one such dataset for each surrogate model. In practice, we can train the surrogate networks, $\{\texttt{MLP}_r\}_{r=1}^{R}$ in parallel to the target network; every K gradient steps optimizing Eqn. 1, we optimize Eqn. 9 to completion for each r, allowing it to "follow" shifts in the target neural model. We found empirically



(a) Random:1 **(b) Random:2** **(c) Random:3** **(d) Random:4** **(e) Fixed:1** **(f) Fixed:2** **(g) Fixed:3** **(h) Fixed:4**

(i) Effect of Deterministism on Surrogate Quality **(j) Effect of Data Augmentation on Surrogate Quality**

Figure 2: (a-d) Decision trees using randomized training; (e-h) decision trees using determinitic training; (i) Comparison of the mean squared error (MSE) between surrogate predictions and the true average path lengths over a training run. Critically, randomized training introduces larger error. (j) Comparison of MSE with and without data augmentation. Using more sample weights to train the surrogate further reduces prediction error.

that each surrogate is a rather low dimensional transformation and quite cheap to train, requiring only a few layers.

The ability of each surrogate to stay faithful to $\Omega_r^{\texttt{regional}}(\cdot)$ is a function of many factors. Wu et al. [29] used a fairly simple strategy for training a surrogate and found it sufficient; we find that especially when there are multiple surrogates to be maintained, sophistication is needed to keep the gradients accurate and the variances low. We describe these innovations in the next section.

4.3 Innovations for Optimization Stability

Even when training only one surrogat (for global tree regularization), we found that in certain settings, the surrogate was unable to accurately predict the APL, causing regularization to fail. Repeated runs also often found very different minima. These issues were only exacerbated when training multiple surrogates. Below, we list optimization innovations that proved to be essential to stabilize training, identify consistent minima, and get good APL prediction—all of which enabled robust regional tree regularization.

Data augmentation makes for a more robust surrogate. Especially for regional explanations, relatively small changes in the underlying model can mean large changes for the pattern in a specific region. As such, the surrogates need to be retrained frequently (e.g. every 50 gradient steps). The practice from Wu et al. [29] of computing the true APL for a dataset $\mathcal{D}^{\theta}$ of the most recent θ is insufficient to learn the mapping from a thousand-dimensional weight vector to the APL. Using stale (very old) θ, however, would result in an APL model that is focused on irrelevant parameter settings. We supplement the recent weights with randomly sampled weight vectors from the convex hull defined by the recent weights. Specifically, to generate a new θ, we sample from a Dirichlet distribution with J categories and form a new parameter as a convex combination of the elements in $\mathcal{D}^{\theta}$. For each of these samples, we compute its true APL to train the surrogate. Fig. 2j shows how this reduces noise in predictions.

Decision trees should be trained deterministically. CART is a common algorithm to train a decision tree. However, it has poor complexity in the number of features as it enumerates over all unique values per dimension. To scale efficiently, many open-source implementations (e.g. Scikit-Learn [24]) randomly sample a small subset of features. As such, independent training instances can lead to different decision trees of varying APL. This is especially a problem if the training dataset is sparse. For tree regularization, variance in APL leads to difficulty in training the surrogate model, since the function from model parameters to APL is no longer many-to-one. The error is compounded when there are many surrogates. To remedy this, we fix the random seed that governs the choice of features. Fig. 2 shows the high variance of decision boundaries from a randomized treatment of fitting decision trees (a-d) on a very sparsely sampled data set, leading to higher error in surrogate predictions (i). Setting the seed removes this variance.

Decision trees should be pruned. Given a dataset, $\mathcal{D}$, even with a fixed seed, there are many decision trees that can fit $\mathcal{D}$. One can always add additional subtrees that predict the same label as the parent node, thereby not effecting performance. This invariance again introduces difficulty in learning a surrogate model. To remedy this, we use *reduced error pruning*, which removes any subtree that does not effect performance as measured on a portion of $\mathcal{D}$ not used in TrainTree. Note that line 4 in Alg. 1 is not in the original tree regularization algorithm. Intuitively, pruning collapses the set of possible trees describing a single classifier to a singleton.

A large learning rate will lead to thrashing. As mentioned before, with many regions, small changes in the deep model can already have large effects on a region. If the learning rate is fast, each gradient step can lead to a dramatically different decision boundary than the previous. Thus, the function that each surrogate must learn is no longer continuous. Empirically, we found large learning rates to lead to *thrashing*, or oscillating between high and low APL where the surrogate is effectively memorizing the APL from the last epoch (with poor generalization to new θ).

4.4 Evaluation Metrics

We wish to compare models with global and regional explanations. However, given $\theta \in \Theta$, $\Omega^{\texttt{regional}}(\theta)$ and $\Omega^{\texttt{global}}(\theta)$ are not directly comparable: subtley, the APL of a global tree is often an overestimate for data points in a single region. To reconcile this, for any globally regularized model, we separately compute $\Omega^{\texttt{regional}}(\theta)$ as an evaluation criterion. In this context, $\Omega^{\texttt{regional}}$ is used only for evaluation; it does not appear in the objective nor training. We do the same for baseline models, L2 regularized models, and unregularized models. From this point on, if we refer to average path length (e.g. Test APL, APL, path length) outside of the objective, we are referring to the evaluation metric, $\Omega^{\texttt{regional}}(\theta)$.

5 DEMONSTRATION ON A TOY EXAMPLE

To build intuition, we present experiments in a new toy setting: We define a ground-truth classification function composed of five rectangles (height of 0.5 and width of 1) in $\mathbb{R}^2$ concatenated along the x-axis to span the domain of $[0, 5]$. The first three rectangles are centered at $y = 0.4$ (shifted slightly downwards) while the remaining two rectangles are centered at $y = 0.6$ (shifted slightly upwards). The training dataset is intended to be sparse, containing only 250 points with the labels of 5% of points randomly flipped to introduce noise and encourage overfitting. In contrast, the test dataset is densely sampled without noise. This is intended to model real-world settings where regional structure is only partially observable from an empirical dataset. It is exactly in these contexts that prior knowledge can be helpful.

Fig. 3 show the learned decision boundary with (c) no regularization, (d) L2 regularization, (e) global tree regularization, and (f) regional tree regularization. As global regularization is restricted to penalizing all data points evenly, it fails to find the happy medium between being too complex or too simple. In other words, increasing the regularization strength quickly causes the target neural model to collapse from a complex nonlinear decision boundary to a single axis-aligned boundary. As shown in (e), this fails to capture any structure imposed by the five rectangles. Similarly, if we increase the strength of L2 regularization even slightly from (d), the model collapses to the trivial solution of predicting entirely one label. Only regional tree regularization (f) is able to model the up-and-down curvature of the true decision function. Knowledge of the regions provides a model with prior information about underlying structure in the data; we should expect that with such information, the model can better regularize itself not to over- or underfit.

We train for 500 epochs with a learning rate or 4e-3, a minibatch size of 32, retrain the surrogate function every epoch (a loop over the full training dataset) and sample 1000 weights from the convex hull each time. Decision trees were trained with $h = 1$. Table 1 compares metrics between the different regularizations: although the regional tree regularization is slightly more complex than global tree regularization, it comes with a large increase in accuracy.

Model	Test Acc.	Test APL
Unregularized	0.8296	17.9490
L2 ($\lambda = 0.001$)	0.8550	16.1130
Global Tree ($\lambda = 1$)	0.8454	6.3398
Regional Tree ($\lambda = 0.1$)	0.9168	10.1223

Table 1: Classification performance on a toy demonstration with varying regularizations. APL on a test set is computed as in Eqn. 4 (averaged over the five regions).

6 RESULTS ON BENCHMARKS

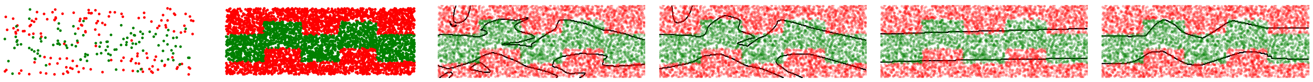
We apply regional tree regularization to a suite of four popular machine learning datasets from UC Irvine repository [5]. We briefly provide context for each dataset and show results comparing the regularization methods in effectiveness. We choose a generic method for defining regions to showcase the wide applicability of regional regularization: we use $\mathcal{D}$ to fit a k-means clustering model with $k = 5$. Each example $\mathbf{x}_n \in \mathcal{D}$ is then assigned a number, $s_n \in \{1, 2, 3, 4, 5\}$. We define $X_r = \{\mathbf{x}_n | s_n = r\} \subseteq \mathcal{X}^P$.

(1) *Bank Marketing* (Bank): 45,211 rows collected from marketing campaigns for a bank [22]. $\mathbf{x}_n$ has 17 features describing a recipient of the campaign (age, education, etc). There is one binary ouput indicating whether the recipient subscribed.
(2) *MAGIC Gamma Telescope* (Gamma): 19,020 samples from a simulator of high energy Gamma particles in an Cherenkov telescope. There are 11 input features that describe afterimages of photon pulses, and one binary output discriminating between signal and background.
(3) *Adult Income* (Adult): 48,842 data points with 14 input features (age, sex, etc.), and a binary output indicating if an individual's income exceeds $50,000 per year [13].
(4) *Wine Quality* (Wine): 4,898 examples describing white wine from Portugal. Each row has a quality score from 0 to 10 and eleven input variables based on physicochemical tests for acidity, sugar, pH, etc. We binarize the target where a positive label indicates a score of at least 5.

In each dataset, the target neural model is trained for 500 epochs with 1e-4 learning rate using Adam [11] and a minibatch size of 128. We train under 20 different λ between 0.0001 and 10.0. We do not do early stopping to preserve overfitting effects. We use 250 samples from the convex hull and retrain every 50 gradient steps. We set $C = 25$ for Wine and $C = 100$ otherwise.

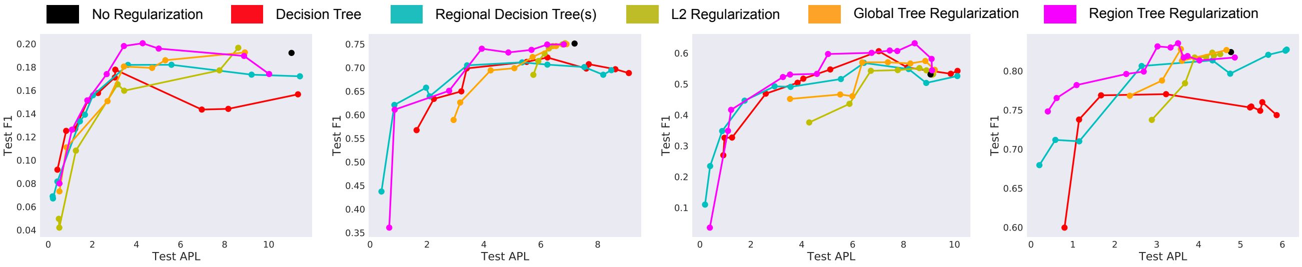
Fig. 4 (a-d) compare L2, global tree, and regional tree regularization with varying strengths. The points plotted show minima from 3 independent runs. We include three baselines: an unregularized model, a decision tree trained on $\mathcal{D}$ and, a set of trees with one for each region (we call this: regional decision tree). For baseline trees, we vary h where a higher h is a more regularized model.

Some patterns are apparent. First, an unregularized model (black) does poorly due to overfitting to a complex decision boundary, as the training dataset is relatively small for an over-parameterized neural network. Second, we find that L2 is not a desirable regularizer for simulatability as it is unable to find many minima in the low APL region (see Gamma, Adult, and Wine under roughly 5 APL). Any increase in regularization strength quickly causes the target neural model to decay to an F1 score of 0, in other words, one



(a) Training set (b) Test set (c) Unregularized (d) L2 (e) Global Tree (f) Regional Tree

Figure 3: Synthetic data with a sparse training set (a) and a dense test set (b). Due to sparsity, the division of five rectangles is not trivial to uncover from (a). (c-f) show contours of decision functions learned with varying regularizations and strengths. Only the regional tree regularized model captures the vertical structure of the five regions, leading to high accuracy.



(a) Bank (b) Gamma (c) Adult (d) Wine

Figure 4: (a-d) Comparison of regularization methods (L2, global tree, regional tree) on four datasets from the UCI repository. Each subfigure plots the average APL over 5 regions (computed on a held-out test set) against the test F1 score. The ideal model is with high accuracy and low APL i.e. the upper left diagonal of each plot. In each setting, regional tree regularized models are able to find more low APL minima than global explanations and consistently achieves the highest performance at low APL. In contrast, the performance of global tree and L2 regularization quickly decays as the regularization strength increases.

that predict a single label. We see similar behavior with global tree regularization, suggesting that finding low complexity minima is challenging under global constraints.

Thirdly, regional tree regularization achieves the highest test accuracy in all datasets. We find that in the lower APL area, regional explanations surpasses global explanations in performance. For example, in Bank, Gamma, Adult, and Wine, we can see this at 3-6, 4-7, 5-8, 3-4 APL respectively. This suggests, like in the toy example, that it is easier to regularize explicitly defined groups rather than the entire input space as a whole. In fact, unlike global regularization, models constrained regionally are able to reach a wealth of minima in the low APL area. Lastly, we note that with high regularization strengths, regional tree regularization converges in performance with regional decision trees, which is sensible as the neural network prioritizes distillation over performance.

7 CASE STUDIES

Now, we consider two real-world use cases: predicting interventions in critical care and predicting medication usage for HIV treatment.

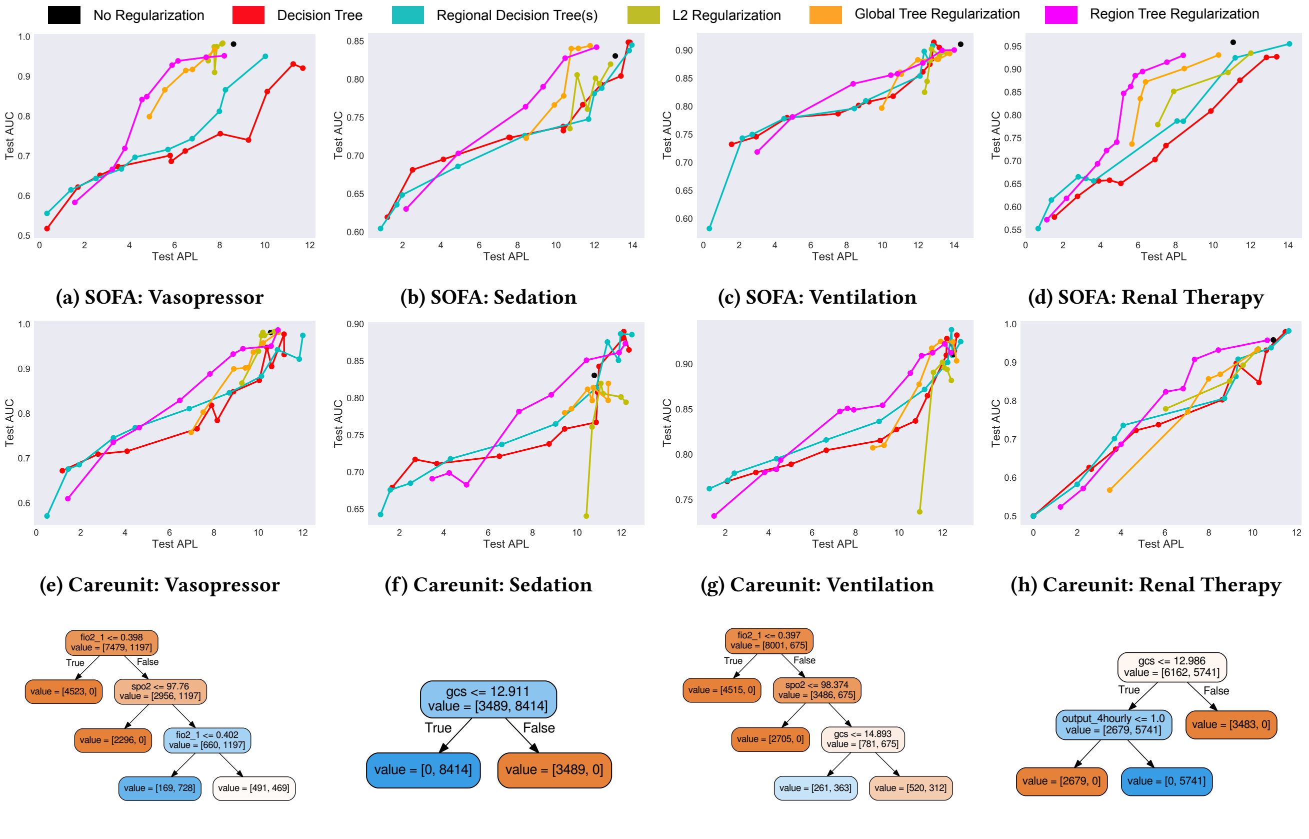
7.1 Critical Care

We study 11,786 intensive care unit (ICU) patients from the MIMIC III dataset [9]. We ignore the temporal dimension, resulting in a dataset $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$ with $P = 35$ input features, and $Q = 4$ binary outcomes. $\mathbf{x}_n$ measures continuous features such as respiration rate (RR), blood oxygen levels (paO$_2$), fluid levels, and more. $\mathbf{y}_n$ detail if vassopressin, sedation, mechanical ventilation, or renal replacement therapy was applied, respectively. Models are trained to predict all output dimensions concurrently from one shared embedding. We discard patients without a recorded careunit. This leaves $6,313$ unique patients with $N = 86,441$ total measurements. We use a 80-10-10 split for training, validation, and test sets, respectively. We will refer to this dataset as *Critical Care*. We set $C = N$. We first describe a few details and then discuss results.

APL for multiple outputs. Previous datasets had only 1 binary output while Critical Care has 5. Fortunately, the definition of APL generalizes: compute the APL for each output dimension (using Eqn. 4), and take the sum as the measure of complexity. Note that this requires fitting $Q \times R$ decision trees.

Defining regions. We explore two methods of defining regions in Critical Care, both of which suggested by ICU physicians. The first defines three regions by sequential organ failure assessment (SOFA), a summary statistic that has historically been used for predicting ICU mortality. Using $\mathcal{D}$, the groups are defined by more than one standard deviation below the mean, one standard deviation from the mean, and more than one standard deviation above the mean. Intuitively, each group should encapsulate a very different type of patient. The second method clusters patients by the his/her careunit into five groups: MICU (medical), SICU (surgical), TSICU (trauma surgical), CCU (cardiac non-surgical), and CSRU (cardiac surgical). Again, patients who undergo surgery should behave differently than those with less-invasive operations.

Regularization results. Fig. 5 compares different regularization schemes against baseline models for SOFA regions (a-d) and careunit regions (e-h). Overall, the patterns we discussed in the UCI datasets are consistent in this application. We especially highlight the inability (across the board) of global explanation to find low



(a) SOFA: Vasopressor

(b) SOFA: Sedation

(c) SOFA: Ventilation

(d) SOFA: Renal Therapy

(e) Careunit: Vasopressor

(f) Careunit: Sedation

(g) Careunit: Ventilation

(h) Careunit: Renal Therapy

(i) Low SOFA: Ventilation

(j) High SOFA: Ventilation

(k) Low SOFA: Sedation

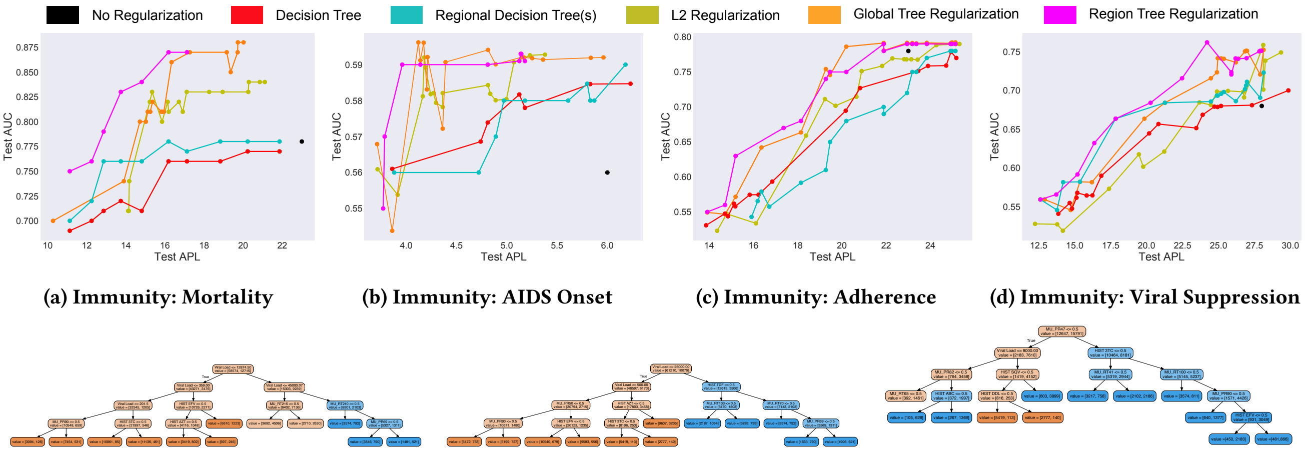
(l) High SOFA: Sedation

Figure 5: Comparison of regularization methods on the Critical Care dataset. Each output represents a form of medication given in the ICU (e.g. vasopressor, sedation, mechanical ventilation, and renal replacement therapy). Each subfigure compares APL and test accuracy. (a-d) compute APL based on three regions defined using SOFA scores; (e-h) instead, compute APL on five regions, one for each careunit (e.g. medical ICU vs. surgical ICU). In each set of experiments, regional tree regularized finds the best performing models at low complexity. Finally, (i-l) show distilled decision trees (split by SOFA) that best approximate a regionally regularized target neural model with a low APL and good test accuracy. As confirmed by a physician in the ICU, distilled trees are simulatable and capture statistical nuances specific to a region.

complexity solutions. For example, in Fig. 5 (a,c,e), the minima from global constraints stay very close to the unregularized minima. In other cases (f, g), global regularization finds very poor optima: reaching low accuracy with high APL. In contrast, region regularization consistently finds a good compromise between complexity and performance. In each subfigure, we can point to a span of APL at which the pink curve is much higher than all others. These results are from three runs, each with 20 different strengths.

Distilled decision trees. A consequence of tree regularization is that every minima is associated with a set of trained trees. We can extract the trees that best approximate the target neural model, and rely on it for explanation. Fig. 5 (i,j) show an example of two trees predicting ventilation plucked from a low APL - high AUC minima of a regional tree regularized model. We note that the composition of the trees are very different, suggesting that they each capture a decision function biased to a region. Moreover, we can see that while Fig. 5 (i) mostly predicts 0, Fig. 5 (j) mostly predicts 1; this agrees with our intuition that SOFA scores are correlated with risk of mortality. Fig. 5 (k,l) show similar findings for sedation. If we were to capture this behavior with a single decision tree, we would either lose granularity or be left with a very large tree.

Feedback from physicians. We presented a set of 9 distilled trees from regional tree regularized models (1 for each output and SOFA region) to an expert intensivist for interpretation. Broadly, he found the regions beneficial as it allowed him to connect the model to his cognitive categories of patients—including those unlikely to need interventions. He verified that for predicting ventilation, GCS (mental status) should have been a key factor, and for predicting vasopressor use, the logic supported cases when vasopressors would likely be used versus other interventions (e.g. fluids if urine output is low). He was also able to make requests: for example, he asked if the effect of oxygen could have been a higher branch in the tree to better understand its effects on ventilation choices, and, noticing the similarities between the sedation and ventilation trees, pointed out that they were correlated and suggested defining new regions by both SOFA and ventilation status.



(a) Immunity: Mortality

(b) Immunity: AIDS Onset

(c) Immunity: Adherence

(d) Immunity: Viral Suppression

(e) High Immunity: Mortality

(f) Mid Immunity: Mortality

(g) Low Immunity: Mortality

Figure 6: Comparison of regularization methods on 15 output dimensions of the HIV dataset (4 of which are shown). Each subfigure compares APL and test accuracy. Subfigures (a-d) base the metric on four regions corresponding to the level of immunosuppression (abbreviated to immunity) at baseline (e.g. <200 cells/mm^3). Subfigures (e-g) show distilled decision trees (split by degrees of immunity) that best approximate a regionally regularized target neural model with a low APL.

We highlight that this kind of reasoning about what the model is learning and how it can be improved is very valuable. Very few notions of interpretability in deep models offer the level of granularity and simulatability as regional tree explanations do.

7.2 HIV

We study 53,236 patients with HIV from the EuResist Integrated Database [30]. Each input $\mathbf{x}_n$ contains 40 features, including blood counts, viral load measurements, and lab results. Each output $\mathbf{y}_n$ has 15 binary labels, including whether a therapy was successful in reducing viral load, and if therapy caused CD4$^+$ blood cell counts to drop to dangerous levels. We only consider those patients for whom we know their degree of immunosuppression in terms of CD4$^+$ count at baseline. All other settings are as in Critical Care.

Defining regions in HIV. We define regions based on the advice of medical experts. This is performed using a patient's degree of immunosuppression at baseline (known as CDC staging). These groups are defined as: <200 cells/mm^3, 200 - 300 cells/mm^3, 300 - 500 cells/mm^3 and >500 cells/mm^3 [23]. This choice of regions should characterize patients based on the initial severity of their infection; the lower the initial cell count, the more severe the infection.

Regularization results. Fig. 6 compares different regularization schemes against baseline models across levels of immunosuppression. Overall, regional tree regularization produces more accurate predictions and provides simpler explanations across all outputs. In particular, for the case of predicting patient mortality in Fig 6a, we tend to find more suitable optima across different patient groupings and can provide better regional explanations for these patients as a result. Here, we observe that patients with lower levels of immunosuppression tend to have lower risk of mortality. We also observe that patients with lower immunity at baseline are more likely to progress to AIDS. Similar inferences can be made for the other outputs. In each subfigure, we reiterate that there is a span of APL at which the pink curve is much higher than all others.

Distilled decision trees. We extract decision trees that approximate the target model for multiple minima and use these as explanations. Fig 6 (e-g) show three trees where we have low APL and high AUC minima from a regional tree regularized model. Again, the trees look significantly different based on the decision function in a particular region. In particular, we observe that lower levels of immunity at baseline are associated with higher viral loads (lower viral suppression) and higher risk of mortality.

Feedback from physicians. The trees were shown to a physician specializing in HIV treatment. He was able to simulate the model's logic, and confirmed our observations about relationships between viral loads and mortality. In addition, he noted that when patients have lower baseline immunity, the trees for mortality contain several more drugs. This is consistent with medical knowledge, since patients with lower immunity tend to have more severe infections, and require more aggressive therapies to combat drug resistance.

8 DISCUSSION

We discuss a few observations about the proposed method.

The most effective minima are found in the low APL, high AUC regime. The ideal model is one that is highly performant and simulatable. This translates to high F1/AUC scores near medium APL. Too large of an APL would be hard for an expert to understand. Too small of an APL would be too restrictive, resulting in no benefit from using a deep model. Across all experiments, we see that region regularization is more adept at finding low APL and high AUC minima than any global regularization method.

Global and local regularization are two extreme forms of regional regularization. If $R = 1$, the full training dataset is contained in a single region, enforcing global explainability. If $R = N$, then every data point $\mathbf{x}_n \in \mathcal{D}$ has its own region i.e. local explainability.

Regularized deep models outperform decision trees. If we compare regional tree regularized deep models and regional decision trees, the former reach much higher accuracy at equal simulatability.

	Bank	Gamma	Adult	Wine	Crit. Care	HIV
Fidelity	0.892	0.881	0.910	0.876	0.900	0.897

Table 2: Fidelity is the percentage of examples on which the prediction made by a tree agrees with the deep model [4].

Regional tree regularization produces regionally faithful decision trees. Table 2 shows the fidelity of a deep model to its distilled decision tree. A score of 1.0 indicates that the deep model and decision tree learned the same decision function. With a fidelity of 89%, the distilled tree is trustworthy in a majority of cases, but can take advantage of deep nonlinearity with difficult examples.

Regional tree regularization is not computationally expensive. Over 100 trials on Critical Care, an L2 model takes 2.393 ± 0.258 sec. per epoch; a global tree model takes 5.903 ± 0.452 sec. and 21.422 ± 0.619 sec. to (1) sample 1000 convex samples, (2) compute APL for $\mathcal{D}^{\theta}$, (3) train a surrogate model for 100 epochs; a regional tree model takes 6.603 ± 0.271 sec. and 39.878 ± 0.512 sec. for (1), (2), and training 5 surrogates. The increase in base cost is due to the extra forward pass through R surrogate models to predict APL in the objective. The surrogate cost(s) are customizable depending on the size of $\mathcal{D}^{\theta}$, the number of training epochs, and the frequency of re-training. If $R >> 0$, we need not re-train each surrogate. The choice of which regions to prioritize can be solved as a bandit problem.

Distilled decision trees are interpretable by domain experts. We separately asked physicians in Critical Care and HIV to analyze the distilled decision trees from regional regularization. They were able to quickly understand the learned decision function per region, suggest improvements, and verify the logic.

Optimizing surrogates is much faster and more stable than gradient-free methods. We tried alternative optimization methods that do not require differentiating through training a decision tree: (1) estimate gradients by perturbing inputs, (2) search algorithms like Nelder-Mead. However, we found these methods to either be unreasonably expensive, or easily stuck in local minima.

9 CONCLUSION

Interpretability is a bottleneck preventing widespread acceptance of deep learning. We propose a regularizer for human-simulatability that adds prior knowledge partitioning the input space into regions. We show the effectiveness of regional tree regularization in learning accurate deep models for healthcare that clinicians can understand.

REFERENCES

[1] Dan Amir and Ofra Amir. 2018. HIGHLIGHTS: Summarizing Agent Behavior to People. In *Proc. of the 17th International conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

[2] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. 2015. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542* (2015).

[3] Jonathan H Chen, Steven M Asch, et al. 2017. Machine learning and prediction in medicine-beyond the peak of inflated expectations. *N Engl J Med* 376, 26 (2017), 2507–2509.

[4] Mark Craven and Jude W Shavlik. 1996. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*. 24–30.

[5] Dua Dheeru and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[6] Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a Neural Network Into a Soft Decision Tree. *arXiv preprint arXiv:1711.09784* (2017).

[7] Marzyeh Ghassemi, Mike Wu, Michael C Hughes, Peter Szolovits, and Finale Doshi-Velez. 2017. Predicting intervention onset in the ICU with switching state space models. *AMIA Summits on Translational Science Proceedings* 2017 (2017), 82.

[8] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama* 316, 22 (2016), 2402–2410.

[9] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[10] Been Kim, Cynthia Rudin, and Julie A Shah. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*. 1952–1960.

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

[12] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017).

[13] Ron Kohavi. 1996. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid.. In *KDD*, Vol. 96. Citeseer, 202–207.

[14] Samantha Krening, Brent Harrison, Karen M Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. 2017. Learning from explanations using sentiment and advice in RL. *IEEE Transactions on Cognitive and Developmental Systems* 9, 1 (2017), 44–55.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[16] Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016).

[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[18] Tim Miller. 2018. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* (2018).

[19] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. 2016. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports* 6 (2016), 26094.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[21] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June* 20, 14 (2015), 5.

[22] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[23] World Health Organization et al. 2005. *Interim WHO clinical staging of HVI/AIDS and HIV/AIDS case definitions for surveillance: African Region.* Technical Report. Geneva: World Health Organization.

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.

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

[27] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Why did you say that? *arXiv preprint arXiv:1611.07450* (2016).

[28] Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. 2016. Programs as Black-Box Explanations. *arXiv preprint arXiv:1611.07579* (2016).

[29] Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. 2017. Beyond Sparsity: Tree Regularization of Deep Models for Interpretability. *arXiv preprint arXiv:1711.06178* (2017).

[30] Maurizio Zazzi, Rolf Kaiser, A Sönnerborg, Daniel Struck, Andre Altmann, Mattia Prosperi, M Rosen-Zvi, A Petroczi, Y Peres, E Schülter, et al. 2011. Prediction of response to antiretroviral therapy by human experts and by the EuResist data-driven expert system (the EVE study). *HIV medicine* 12, 4 (2011), 211–218.