Regional Tree Regularization for Interpretability in Black Box Models

Mike Wu
Stanford University
Stanford, US
wumike@stanford.edu

Sonali Parbhoo
University of Basel
Basel, Switzerland
sonali.parbhoo@unibas.ch

Michael C. Hughes
Tufts University
Medford, US
michael.hughes@tufts.edu

Ryan Kindle
Massachusetts General Hospital
Boston, US

Leo Anthony Celi
MIT
Cambridge, US
email address

Maurizio Zazzi
University of Siena
Siena, Italy
maurizio.zazzi@unisi.it

Volker Roth
University of Basel
Basel, Switzerland
volker.roth@unibas.ch

Finale Doshi-Velez
Harvard University SEAS
Cambridge, US
finale@seas.harvard.edu

***Abstract*—The lack of interpretability remains a barrier to the adoption of deep neural networks. Recently, tree regularization has been proposed to encourage deep neural networks to resemble compact, axis-aligned decision trees without significant compromises in accuracy. However, it may be unreasonable to expect that a single tree can predict well across all possible inputs. In this work, we propose *regional* tree regularization, which encourages a deep model to be well-approximated by several separate decision trees specific to predefined regions of the input space. Practitioners can define regions based on domain knowledge of contexts where different decision-making logic is needed. Across many datasets, our approach delivers more accurate predictions than simply training separate decision trees for each region, while producing simpler explanations than other neural net regularization schemes without sacrificing predictive power. Two healthcare case studies in critical care and HIV demonstrate how experts can improve understanding of deep models via our approach.**

***Index Terms*—Tree Regularization, Interpretability, Explainable AI, Deep Neural Networks**

I. Introduction

Deep models have become the state-of-the-art in applications ranging from image classification [1] to game playing [2], and are poised to advance prediction in real-world domains such as healthcare [3]–[5]. However, understanding when a model's outputs can be trusted and how the model might be improved remains a challenge. [6] discuss how these challenges inhibit the adoption of deep models in clinical settings. Without interpretability, humans are unable to incorporate their domain knowledge and effectively audit predictions.

As such, many efforts have been devoted to extracting explanation from deep models post-hoc. Prior work has focused on two opposing regimes. Works on *global* explanation (e.g. [7], [8]) return a single explanation for the *entire* model. Unfortunately, if the explanation is simple enough to be understandable, then it is unlikely to be faithful to the deep model across all inputs. In contrast, works on *local* explanation (e.g. [9]–[11]) seek to explain individual predictions for a specific input feature vector. These explanations lack generality, as isolated glimpses to the model's behavior can fail to capture larger patterns. Perhaps more troubling, local approaches have trouble indicating whether the same logic revealed for an input x can be used for nearby inputs x'. This ambiguity can lead to mistaken assumptions and poor decisions.

In this work, we consider a middle-ground: *regional* explanations that constrain the model independently across a partitioning of the input space. This form of explanation is consistent with those of humans, whose models are typically context-dependent [12]. 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 interpretable allows the deep model more flexibility than a global constraint, while still revealing prediction logic that can generalize to nearby inputs. Having experts explicitly define regions offers an elegant way to add prior knowledge.

We focus on (regionally) *human-simulatable* explanation [13]. Simulatable explanations allows humans to, in reasonable time, combine inputs and explanation to produce outputs, forming a foundation for auditing and correcting predictions. However, optimizing for simulatable explanations across many regions poses a difficult technical challenge, facing issues with differentiability, efficiency, and a delicate balance of constraints between regions of varying size and complexity. In this paper, we describe a computationally tractable and reliable approach to do so. Specifically, we (1) show how to jointly train a deep model that both has high accuracy and is regionally simulatable, (2) specify a family of novel regularizers, (3) introduce inference innovations for stability in optimization, and (4) demonstrate that we achieve comparable performance to more complex models while learning a much simpler decision function.

II. Related Work

Global Interpretability Given a *trained* black box model, many approaches exist to extract what the model has learned. Works such as [14] expose the features a representation encodes but not the logic. [15], [16] provide an informative

set of examples that summarize the system. Model distillation compress a source network into a smaller target neural network [17]. However, even a small neural model may not be interpretable. Closest to our work, [7] regularize a neural model to behave like a simple decision tree.

Local Interpretability In contrast, local approaches provide explanation for a specific input. [9] show that 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, [18] and [19] output a simple program or an influence function, respectively. Other approaches have used input gradients (which can be thought of as infinitesimal perturbations) to characterize the local space [10], [20]. However, the notion of a local region in these works is both very small and often implicit; it does not match with human notions of contexts [12].

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. [7], [11] 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, [21] 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.

III. Background and Notation

We consider supervised learning tasks given a dataset of N labeled examples, $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, with continuous inputs $\mathbf{x} \in \mathcal{X}^P$ and discrete[1] outputs $\mathbf{y} \in \mathcal{Y}^Q$.

Multi-Layer Perceptrons We focus on multi-layer perceptrons (MLPs) as our representative deep neural network in this work; that said, our ideas can be easily applied to other architectures including recurrent and convolutional networks. The MLP has a vector of parameters θ such that the prediction for $\mathbf{y}_n$ is given by some function $\hat{\mathbf{y}}_n = f(\mathbf{x}_n; \theta)$. The parameters θ are trained to minimize an objective

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

In this work, we assume y_{nq} is binary and use the logistic loss for $\mathcal{L}(\cdot)$. The function $\Omega(\theta)$ represents a regularization penalty, with scalar strength $\lambda \in \mathbb{R}^+$. Common regularizers include the L_1 or L_2 norm of θ. In the following, we shall refer to the predictor $f(\cdot; \theta)$ as our *target neural model*.

Global Tree Regularization [7] introduce a regularization term that penalizes models for being hard to (human-)simulate where simulatability is measured by the "size" (or complexity) of the decision tree that best approximates the target neural model. They define tree complexity as the *average decision path length* (APL), or the expected number of decision nodes that must be touched to make a prediction:

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

where f is the target neural model. The AvgPathLength procedure is defined in Alg. 1, where the subroutine TrainTree refers to any algorithm to fit a *sufficiently* faithful decision tree given input and output pairs (e.g. CART [22]). PruneTree refers to removing "unnecessary" subtrees that do not effect prediction. Note that a disjoint (or validation) portion of the dataset is reserved for measuring pruning error. GetDepth is a subroutine that returns the depth of the leaf node associated with an input example $\mathbf{x}_n$ i.e. it is the length of the trajectory from root to leaf.

Algorithm 1 AvgPathLength (Average Path Length [7])

Require:
$f(\cdot, \theta)$: discrete prediction function, with parameters θ
$\{\mathbf{x}_i\}_{i=1}^N$: a set of N input examples
N_{train}: number of examples to use for training
h: minimum number of samples to define a leaf node

function AvgPathLength($\{\mathbf{x}_i\}_{i=1}^N, f, h$)
  $\hat{\mathbf{y}}_i = f(\mathbf{x}_i, \theta)$, $\forall i \in \{1, 2, \dots N\}$
  $T = \textsc{TrainTree}(\{\mathbf{x}_i, \hat{\mathbf{y}}_i\}_{i=1}^{N_{\text{train}}}, h)$
  $T = \textsc{PruneTree}(T, \{\mathbf{x}_i, \hat{\mathbf{y}}_i\}_{i=N_{\text{train}}}^{N})$
  return mean($\{\textsc{GetDepth}(T, \mathbf{x}_i)\}_{i=1}^N$)

However, TrainTree is not differentiable, making the optimization of Eqn. 2 challenging. Thus, [7] introduce a surrogate regularizer $\hat{\Omega}^{\texttt{global}}(\theta)$, which maps the parameter vector θ from the target neural model to an estimate of the APL. In practice, $\hat{\Omega}^{\texttt{global}}(\theta)$ is a small neural network. [7] refers to this as the *surrogate model*.

Training the surrogate model is a supervised problem. First, [7] 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. Next, they optimize the following objective:

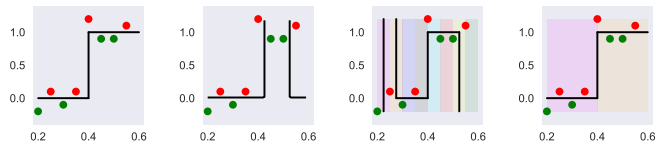
$$\arg\min_{\phi \in \Phi} \textstyle\sum_{j=1}^{J} (\Omega^{\texttt{global}}(\theta_j) - \hat{\Omega}^{\texttt{global}}(\theta_j; \phi))^2 \quad (3)$$

Critically, the optimal parameters of the surrogate model, $\phi \in \Phi$, depend on the value of θ. Every J gradient steps in training the target neural model, they freeze θ and optimize ϕ to completion. This represents updating the mapping $\hat{\Omega}^{\texttt{global}}$, as the target neural model changes during learning.

IV. Regional Faithful Explanations

Global summaries such as [7] face a tough trade-off between human-simulatability and being faithful to the underlying model. Too simple of a summary would no longer describe the predictions of the neural network; too faithful of a summary and it is no longer understandable. To get the best of both worlds, we need a finer-grained definition interpretability for each input. For example, an intensivist may already cognitively consider patients in the surgical intensive care unit (ICU) as different from patients in the cardiac ICU. Analogously,

[1] Extensions to continuous outputs are straightforward.



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

Fig. 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 over-regularized 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.

biologists may be happy with different models for classifying diseases in deciduous versus in coniferous plants. We thus divide the input space into exclusive regions. We assume that this division is available *a priori* via domain knowledge.

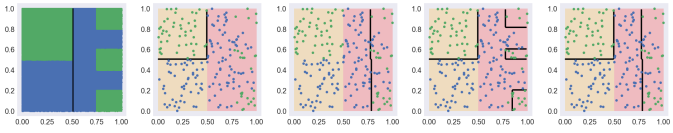
Formally, we assume there are R exclusive regions $\mathcal{X}_1, \ldots \mathcal{X}_R$, where $\cup_{r=1}^{R} \mathcal{X}_r \subseteq \mathcal{X}^P$. We denote the observed dataset belonging to region r as $X_r \triangleq \{x_n : x_n \in \mathcal{X}_r\}$.

The cognitive science literature tells us that people build context-dependent models of the world; they do not expect the same rule to apply in all circumstances [12]. Thus, we shall apply a *regionally-faithful regularization* that encourages the target neural model to be "simple" in *every* region (where a region corresponds to a human context). We emphasize that our regional explanations are distinct from local explanations (e.g. [9]): the latter concerns itself with behavior within an ϵ-ball around a single data point, $\mathbf{x}_n$ and makes no claims about general behavior across data points. In contrast, *regional* explanations are faithful over an entire region $\mathcal{X}_r$.

As a preview, Fig. 1 highlights the distinctions between global, local, and regional tree regularization on a toy dataset where the true decision boundary is divided in half at $x = 0.4$. We see that global explanations (b) lack information about the input space and have to choose from a large set of possible solutions, converging to a different boundary. On the other hand, local explanations (c) produce simple boundaries around each data point but fail to capture global relationships, resulting in a complex overall decision function. Finally, regional explanations (d) over two regions divided at 0.4 share the benefits of (b) and (c), converging to the true boundary.

A. Regional Tree Regularization

We now introduce regional tree regularization, which will require that the target neural model f is well-approximated by a separate compact decision tree in *every* region (trees can modeling nonlinearity while remaining human-simulatable). This is particularly hard to achieve with global tree regularization of [7] as their global APL metric may allow some human-relevant regions to be complex as long as most are



(a) True (b) L_1 (c) L_1 (d) L_0 (e) L_0

Fig. 2: An L_1 penalty on per-region average path lengths can over-penalize compared to the L_0 norm, resulting in an entire region with far too simple predictions. Subplots (b) and (c) show results from two different initializations using the L_1 norm, while (d) and (e) show the same using the L_0 norm.

simple. In contrast, we define our regional tree regularization as follows. First, let the average path length for region r be:

$$\Omega_r^{\texttt{regional}}(\theta) \triangleq \textsc{AvgPathLength}(X_r, f(\cdot)), \qquad (4)$$

which can be computed with Alg. 1 (note that the target network and its parameters θ are the same for all regions r). Next, to ensure that some regions cannot be made simple at the expense of others, we penalize only the most complex region:

$$\Omega^{\texttt{regional}}(\theta) \triangleq \texttt{max}_r(\{\Omega_r^{\texttt{regional}}(\theta)\}_{r=1}^{R}) \qquad (5)$$

in other words, a L_0 norm over $\{\Omega_r\}$. The choice of L_0 norm produces significantly different (and desirable) behavior than if we had simply used, for example, the L_1 norm (or sum) over $\{\Omega_r\}$. Regularizing the sum of Ω_r is equivalent to simply regularizing APL in a global tree that first branches by region. In contrast, as a nonlinear regularizer, L_0 keeps *all* regions simple, while not penalizing regions that are already simple.

We show an example of this effect in Fig. 2: Fig. 2a shows a toy dataset with two regions (split by the black line): the left has a simple decision boundary dividing the region in half; the right has a more complex boundary. Fig. 2b,c then show two minima using L_1 regional tree regularization. In both cases, one of the regions collapses to a trivial decision boundary (predicting all one label) to minimize the overall sum of APLs. On the other hand, since L_0 is sparse, simple regions are not included in the objective, resulting in a more "balanced" regularization between regions (see Fig. 2d,e).

However, gradient descent with Eqn. 5 has several challenges: both Ω_r and the max functions are non-differentiable. In the following, we describe how we address these challenges as well as concerns over optimization stability.

Algorithm 2 SPARSEMAX FOR REGIONAL TREE REG.

Require:
$\hat{\mathbf{\Omega}} = \{\hat{\Omega}_r^{\texttt{regional}}\}_{r=1}^{R}$: APL for each of R regions
function SPARSEMAX($\hat{\mathbf{\Omega}}$)
  Sort $\hat{\mathbf{\Omega}}$ such that $\hat{\mathbf{\Omega}}[i] \geq \hat{\mathbf{\Omega}}[j]$ if $i \geq j$
  $k = \max\{r \in [1, R] | (1 + r\hat{\mathbf{\Omega}}[r]) > \sum_{i \leq r} \hat{\mathbf{\Omega}}[i]\}$
  $\tau = k^{-1}(-1 + \sum_{i \leq k} \hat{\mathbf{\Omega}}[i])$
  $\mathbf{p} = \{p_r\}_{r=1}^{R}$ where $p_r = \max\{\hat{\mathbf{\Omega}}_r - \tau, 0\}$
  return $\mathbf{p}$

B. Gradient-based optimization with SparseMax

Gradient-based optimization of our proposed regularizer in Eqn. 5 is challenging because the $\max$ operator is not differentiable. Further, common differentiable approximations like $\texttt{softmax}$ are dense (include non-zero contributions from all regions), which makes it difficult to focus on the most complex regions as $\max$ does. Instead, we use the recently-proposed SPARSEMAX transformation [23], which can focus on the most problematic regions (setting others to zero contribution) while remaining smooth and differentiable almost everywhere. Intuitively, SPARSEMAX corresponds to a Euclidean projection of an input vector $\hat{\mathbf{\Omega}}$ with R entries (one APL per region) to an R-length vector $\mathbf{p}$ of non-negative entries that sums to one (i.e. the $R-1$-dimensional probability simplex). When the projection lands on a boundary in the simplex (which is likely), then the resulting vector will be sparse. Efficient implementations of this projection are well-known [24] (see Alg. 2), as are Jacobians for automatic differentiation [23]. We refer to using SPARSEMAX as L_0 regional tree regularization.

C. Differentiable Decision-Tree Loss $\hat{\Omega}_r$

The regional APL $\Omega_r(\theta)$ is not differentiable as derivatives cannot flow through CART. To circumvent this, we use a shallow MLP as a *surrogate* loss function $\hat{\Omega}_r^{\texttt{regional}}$: that maps a parameter vector $\theta \in \Theta$ to an *estimate* of $\Omega_r^{\texttt{regional}}(\cdot)$, the APL in region r. Each surrogate $\hat{\Omega}_r$ has its own parameters ϕ_r. The R surrogate models are trained jointly. Specifically, we fit $\hat{\Omega}_r^{\texttt{regional}}(\theta)$ by minimizing a mean squared error loss,

$$\min_{\phi_r} \sum_{j=1}^{J} (\Omega_r^{\texttt{regional}}(\theta_j) - \hat{\Omega}_r^{\texttt{regional}}(\theta_j, \phi_r))^2 \qquad (6)$$

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 over J gradient steps while training the target model $f(\cdot, \theta)$. For R regions, we curate one such dataset for each surrogate model. In practice, we can train the surrogate models in parallel to the target model: every J gradient steps optimizing Eqn. 1, we assemble datasets and optimize Eqn. 6 to completion for each r, allowing them to "follow" shifts in the target model. We found empirically that each surrogate is a low dimensional transformation and cheap to train, requiring only a few layers.

The ability of each surrogate to stay faithful is a function of many factors. [7] 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.

D. Innovations for Optimization Stability

Naively optimizing multiple surrogate networks is a delicate operation. Even when training only one surrogate (for global tree regularization), we found that depending on hyperparameters, the surrogate was unable to accurately predict the APL, causing regularization to fail. Further, repeated runs also often found very different minima, making tree regularization feel unreliable. 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. Without them, tree regularization is not usable at scale.

Experiment	Mean MSE	Max MSE
No data augmentation	0.069	0.987
With data augmentation	0.015	0.298
Non-Deterministic Training	0.116	1.731
Deterministic Training	0.024	0.371

TABLE I: Comparison of the average and max mean squared error (MSE) between surrogate predictions and true average path lengths over a training of 500 epochs. Non-deterministic training and lack of data introduces large errors.

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 [7] 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 from previous epochs, however, would result in a poor surrogate model given outdated information. Thus, we supplement the dataset 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. Table I shows this to reduce noise in predictions.

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.

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 [25]) randomly sample a small subset of features. As such, independent training instances can lead to different decision trees of varying APL. For tree regularization, unexplained variance in APL means difficulty in training the surrogate model, since the

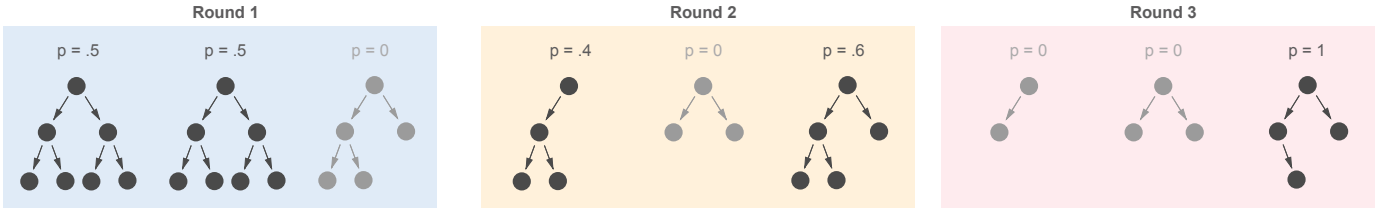
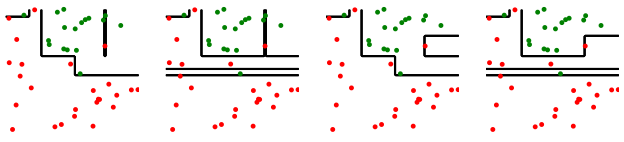
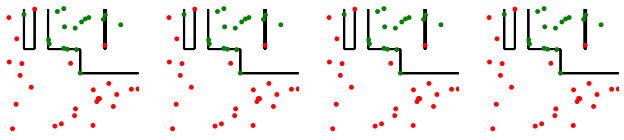


Fig. 3: Demonstration of L_0 regional tree regularization. Each round contains three trees representing three regions. Light gray color indicates regions given 0 probability by `sparsemax`. Over the three rounds, different regions are given priority while other regions are given no weight. The ability to disregard regions of low complexity makes for a smoother optimization.

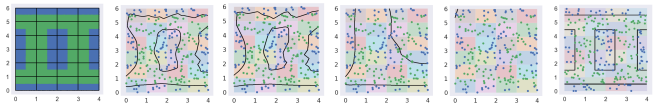


(a) Random:1 (b) Random:2 (c) Random:3 (d) Random:4



(e) Fixed:1 (f) Fixed:2 (g) Fixed:3 (h) Fixed:4

Fig. 4: (a-d) Decision trees using randomized training; (e-h) decision trees using deterministic training.



(a) (b) (c) (d) (e) (f)

Fig. 5: (a) Ground truth decision boundary with 25 regions; green represents positive labels. (b) Minima with no regularization. (c) Minima with no data augmentation. (d) Minima with no pruning or determinism in training trees. (e) Minima with bad learning rate. (f) Minima using optimization innovations. Colored patches represent regions.

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. As an example, Fig. 4 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 (Table. I). Setting the seed removes this variance.

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).

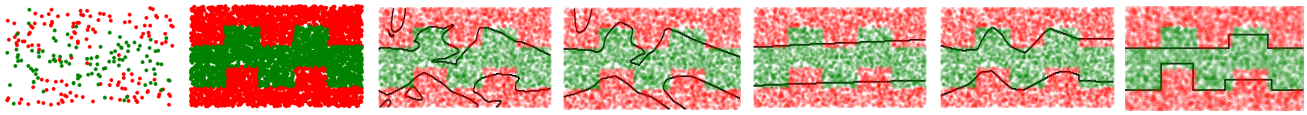
These optimization innovations are crucial for learning with regional tree regularization. Without them, optimization is very unstable, resulting in undesirable minima. Fig. 5 shows a few examples in a synthetic dataset: without data augmentation (c), there are not enough examples to fully train each surrogate, resulting in poor estimates of $\Omega^{\texttt{regional}}$ in which we converge to the same minima as no regularization (b); without pruning and fixing seeds, the path lengths vary due to randomness in fitting a decision tree, which can lead to over- or under-estimating the true APL. As shown in (d), this leads to strange decision boundaries. Finally, (e) shows the effect of large learning rates that leads to thrashing, resulting in a trivial decision boundary in efforts to minimize the loss. Only with the optimization innovations (f), do we converge to a properly regularized decision boundary.

E. 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: subtly, 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, $\Omega^{\texttt{regional}}(\theta)$.

V. Demonstration on a Toy Example

To build intuition, we present experiments in a 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



(a) Training set (b) Test set (c) Unregularized (d) L2 (e) Global Tree (f) L_1 Reg. Tree (g) L_0 Reg. Tree

Fig. 6: 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-g) 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.

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. 6 show the learned decision boundary with (c) no regularization, (d) L2 regularization, (e) global tree regularization, and (f,g) 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,g) is able to model the up-and-down curvature of the true decision function. With high λ, L_0 regional tree regularization produces a more axis-aligned decision boundary than its L_1 equivalent, primarily because we can regularize complex regions more harshly without collapsing simpler regions. Knowledge of the region divisions provides a model with prior information about underlying structure in the data; we should expect that with such information, a regionally regularized model can better prevent itself from over- or underfitting.

We train for 500 epochs with a learning rate of 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 II 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
L_1 Regional Tree ($\lambda = 0.1$)	0.9168	10.1223
L_0 Regional Tree ($\lambda = 0.1$)	0.9308	8.1962

TABLE II: Classification performance on a toy demonstration with varying regularizations. The reported test APL is averaged over APLs in each of the five regions.

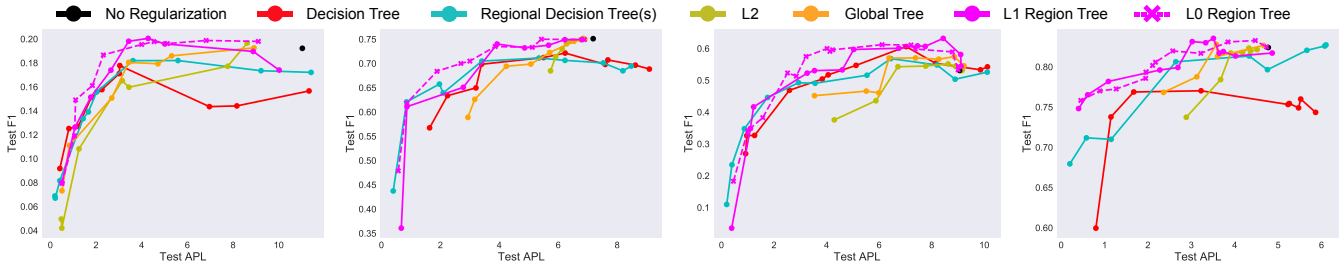
VI. Results on Benchmarks

We now apply regional tree regularization to a suite of four popular machine learning datasets from UC Irvine repository [26]. 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 [27]. $\mathbf{x}_n$ has 17 features describing a recipient of the campaign (age, education, etc). There is one binary output 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 [28].
4) *Wine Quality* (Wine): 4,898 examples describing wine from Portugal. Each row has a quality score from 0 to 10 and eleven 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.

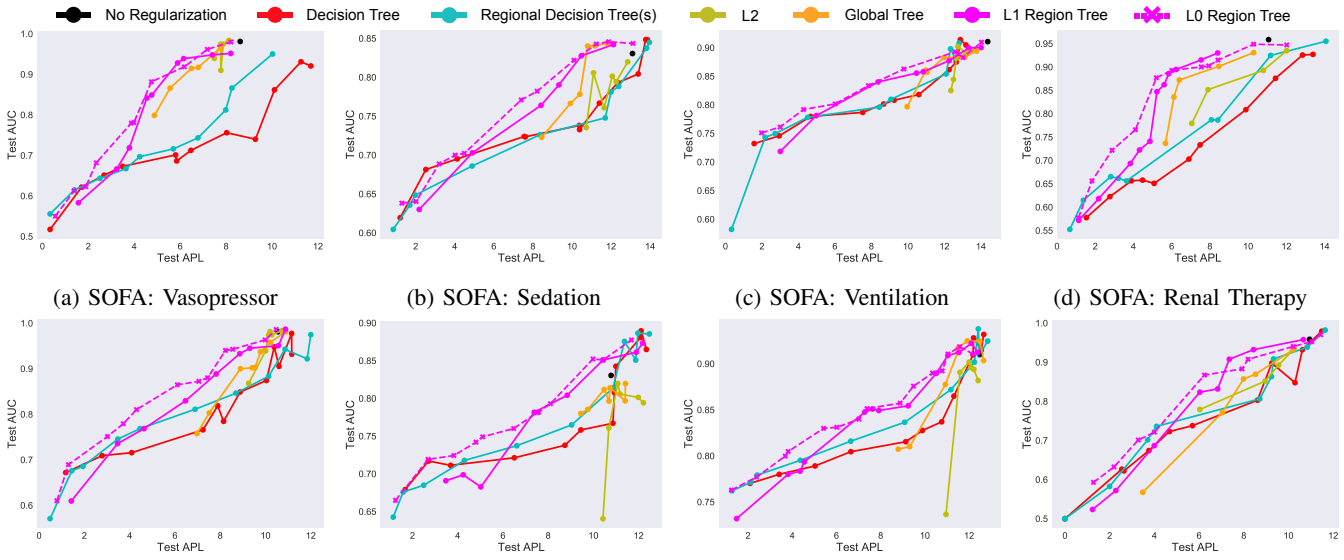
In each dataset, the target neural model is trained for 500 epochs with 1e-4 learning rate using Adam [29] and a minibatch size of 128. We train under 20 different λ between 0.0001 and 10.0. We do not do early stopping to preserve overfitting effects. We use 250 samples from the convex hull and retrain every 50 gradient steps. Fig. 7 (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



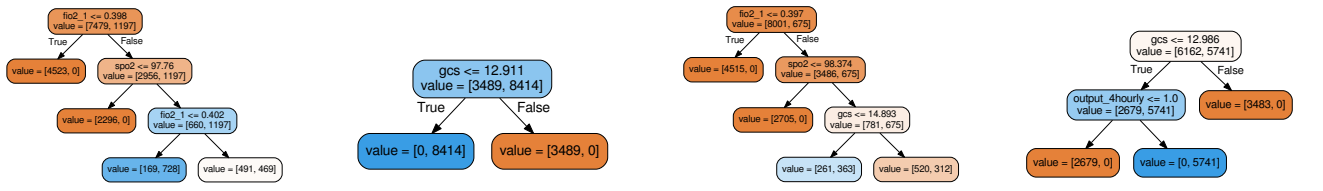
(a) Bank (b) Gamma (c) Adult (d) Wine

Fig. 7: (a-d) Comparison of regularizers (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.



(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

Fig. 8: 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.

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

under global constraints.

Third, 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. Moreover, we note that with high regularization strengths, regional tree regularization mostly converges in performance with regional decision trees, which is sensible as the neural network prioritizes distillation over performance. Finally, again consistent with toy examples, L_0 regional tree regularization finds more performant minima with low to mid APL than its L_1 counterpart across all datasets. We believe this to largely be due to "evenly" regularizing complex and simple regions via sparsity.

VII. Case Studies

we turn to two real-world use cases: predicting interventions in critical care and predicting HIV medication usage.

A. Critical Care

We study 11,786 intensive care unit (ICU) patients from the MIMIC III dataset [30]. 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. $\mathbf{y}_n$ measures if vassopressin, sedation, mechanical ventilation, or renal replacement therapy was applied, respectively (binary label). 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 describe a few details then discuss results.

APL for multiple outputs. Previous datasets had only 1 output dimension while Critical Care has 5. Fortunately, the definition of APL generalizes: compute the APL for each output dimension, and take the sum as the measure of complexity. Note that this requires fitting $Q \times R$ 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. Given a dataset, the regions 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 region 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. 8 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 complexity solutions. For example, in Fig. 8 (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 curves dare much higher than all others. These results are from three runs, each with 20 different strengths. L_0 regional tree reg. in particular (again) dominates the other other methods in minima with low and mid APL.

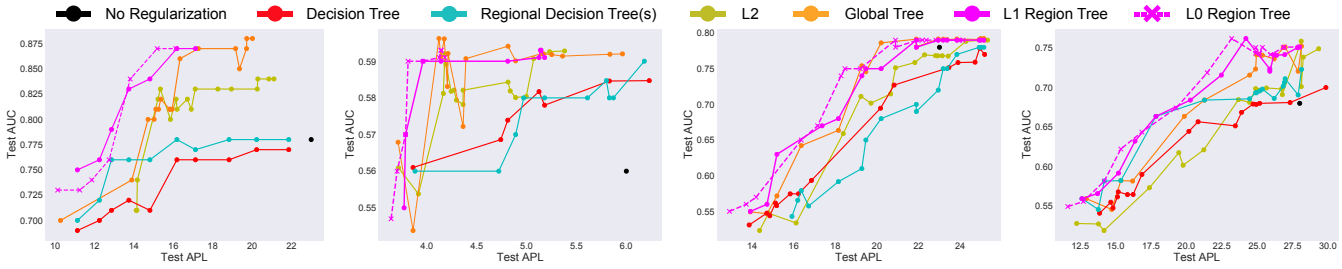
Distilled decision trees. A consequence of tree regularization is that every minima is associated with a set of trained trees. We can extract the trees that best approximate the target neural model, and rely on it for explanation. Fig. 8 (i,j) show an example of two trees predicting ventilation plucked from a low APL - high AUC minima of a regional tree regularized model. We note that the composition of the trees are different, suggesting that they each capture a decision function biased to a region. Moreover, we can see that while Fig. 8 (i) mostly predicts 0, Fig. 8 (j) mostly predicts 1; this agrees with our intuition that SOFA scores are correlated with risk of mortality. Fig. 8 (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.

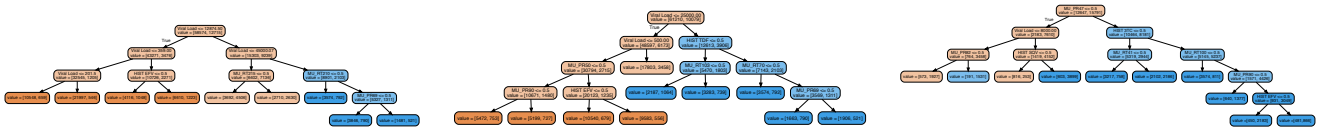
We highlight that reasoning about what the model is learning and how it can be improved is very valuable. Very few notions of interpretability in deep models offer the level of granularity and simulatability as regional tree explanations do.

B. HIV (EuResist)

We study 53,236 patients with HIV from the EuResist Integrated Database [31]. Each input $\mathbf{x}_n$ contains 40 features,



(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

Fig. 9: Comparison of regularizers 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.

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 [32]. 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. 9 compares different regularization schemes against baseline models across levels of immunosuppression. Overall, L_0 regional tree regularization produces more accurate predictions and provides simpler explanations across all outputs. For the case of predicting patient mortality in Fig 9a, 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 dotted 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 9 (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.

VIII. 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 L_0 region regularization is most adept at finding low APL and high AUC minima.

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 trees. Comparing regional tree regularized deep models and regional decision trees, the former reach much higher accuracy at equal APL.

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

TABLE III: Fidelity is the percentage of examples on which the prediction made by a tree agrees with the deep model [33].

Regional tree regularization produces regionally faithful decision trees. Table III shows the fidelity of a deep model to its distilled decision tree. A score of 1.0 indicates that both models learned the same decision function. With a fidelity of 89%, the distilled tree is trustworthy in most 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 is large, we need not re-train each surrogate. The choice of which regions to prioritize can be framed as a bandit problem.

Distilled decision trees are interpretable by domain experts. We 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 based on initialization.

Sparsity over regions is important. We experimented with different "dense" norms: L_1, L_2, and a softmax approximation to L_0, all of which faced issues where regions with simpler decision boundaries a priori were over-regularized to trivial decision functions. Only with L_0 (i.e. `sparsemax`) did we avoid this problem. As a consequence, in toy examples, we observe that `sparsemax` finds minima with more axis-aligned boundaries. In real world studies, we find `sparsemax` to lead to better performance in low/mid APL regimes.

IX. Conclusion

Interpretability is a bottleneck preventing widespread acceptance of deep learning. We propose a novel 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 neural networks for healthcare that clinicians can understand.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[3] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: an unsupervised representation to predict the future of patients from the electronic health records," *Scientific reports*, vol. 6, p. 26094, 2016.

[4] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *Jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

[5] M. Ghassemi, M. Wu, M. C. Hughes, P. Szolovits, and F. Doshi-Velez, "Predicting intervention onset in the icu with switching state space models," *AMIA Summits on Translational Science Proceedings*, vol. 2017, p. 82, 2017.

[6] J. H. Chen, S. M. Asch *et al.*, "Machine learning and prediction in medicine-beyond the peak of inflated expectations," *N Engl J Med*, vol. 376, no. 26, pp. 2507–2509, 2017.

[7] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez, "Beyond sparsity: Tree regularization of deep models for interpretability," *arXiv preprint arXiv:1711.06178*, 2017.

[8] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[10] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?" *arXiv preprint arXiv:1611.07450*, 2016.

[11] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," *arXiv preprint arXiv:1703.03717*, 2017.

[12] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, 2018.

[13] Z. C. Lipton, "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*, 2016.

[14] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google Research Blog. Retrieved June*, vol. 20, no. 14, p. 5, 2015.

[15] D. Amir and O. Amir, "Highlights: Summarizing agent behavior to people," in *Proc. of the 17th International conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2018.

[16] B. Kim, C. Rudin, and J. A. Shah, "The bayesian case model: A generative approach for case-based reasoning and prototype classification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1952–1960.

[17] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," *arXiv preprint arXiv:1711.09784*, 2017.

[18] S. Singh, M. T. Ribeiro, and C. Guestrin, "Programs as black-box explanations," *arXiv preprint arXiv:1611.07579*, 2016.

[19] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," *arXiv preprint arXiv:1703.04730*, 2017.

[20] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[21] S. Krening, B. Harrison, K. M. Feigh, C. L. Isbell, M. Riedl, and A. Thomaz, "Learning from explanations using sentiment and advice in rl," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 1, pp. 44–55, 2017.

[22] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and regression trees*, ser. Wadsworth Statistics/Probability. Chapman and Hall, 1984.

[23] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning*, 2016, pp. 1614–1623.

[24] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," in

Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 272–279.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[26] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[27] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[28] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid." in *KDD*, vol. 96. Citeseer, 1996, pp. 202–207.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[31] M. Zazzi, R. Kaiser, A. Sönnerborg, D. Struck, A. Altmann, M. Prosperi, M. Rosen-Zvi, A. Petroczi, Y. Peres, E. Schülter *et al.*, "Prediction of response to antiretroviral therapy by human experts and by the euresist data-driven expert system (the eve study)," *HIV medicine*, vol. 12, no. 4, pp. 211–218, 2011.

[32] W. H. Organization *et al.*, "Interim who clinical staging of hvi/aids and hiv/aids case definitions for surveillance: African region," Geneva: World Health Organization, Tech. Rep., 2005.

[33] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in neural information processing systems*, 1996, pp. 24–30.