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

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 explanations 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 it to capture patterns across data. However, optimization with many regions becomes 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 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 parent.

Local Interpretability In contrast, these approaches focus on providing explanation for a specific input. Ribeiro et al. [25] 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, Singh et al. [28] and Koh
and Liang [12] output a simple program or an influence function, re-
spectively. 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 Interpretabilities Deep models have many local
optima, some of which may admit more human-simulatable ex-
planations 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 en-
couraged 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
eamples, \(\mathcal{D} = \{x_n, y_n\}_{n=1}^N\), where each example (indexed by \(n\)) has
an input feature vector \(x_n = (x_{n,1}, x_{n,2}, \ldots, x_{n,p}) \in \mathcal{X}^p\) and a target
output vector \(y_n = (y_{n,1}, y_{n,2}, \ldots, y_{n,q}) \in \mathcal{Y}^q\), where \(p, q \in \mathbb{N}\) are
the dimensionalities. We assume each target \(y_{n,q} \in y_n\) is a discrete
dicator 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 architec-
tures including recurrent and convolutional networks. Formally,
a perceptron defines a parameterized function, \(\hat{y}_n = f(x_n; \theta)\),
\(f : \mathcal{X}^p \rightarrow \mathcal{Y}^q\) where \(\hat{y}_n = (\hat{y}_{n,1}, \hat{y}_{n,2}, \ldots, \hat{y}_{n,q})\) are estimates for
the true target \(y_n\). 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 \(\theta\) to minimize the objective,

\[
\arg\min_{\theta \in \Theta} \sum_{n=1}^N \mathcal{L}(y_n, f(x_n; \theta)) + \lambda \Omega(\theta). \quad (1)
\]

For binary targets \(y_{n,q}\), the logistic loss is an effective choice for \(\mathcal{L}(\cdot)\).
The function \(\Omega : \Theta \rightarrow \mathbb{R}\) represents a regularization term, with
scalar strength \(\lambda \in \mathbb{R}^+\). Standard penalty functions include the L1
or L2 norm of \(\Theta\). 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 regulariza-
tion 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 \(x_n \in \mathcal{D}\).

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

4 REGIONAL FAITHFUL EXPLANATIONS

Attempts at global summaries have to tradeoff between being
human-simulatable and being faithful to the underlying model.
In this section, we introduce a more fine-grain definition of inter-
pretability. We assume that the space of inputs \(\mathcal{X}^p\) can be divided
into a set of \(R\) regions \(X_1, \ldots, X_R\) where \(\bigcup_{r=1}^R X_r \subseteq \mathcal{X}^p\) and
each region 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.
Regional Tree Regularization for Interpretability in Black Box Models

In this work, we assume 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 e-ball around a single data point, $x_r \in X^P$. It makes no claims about general behavior across data points. In contrast, regional explanations are faithful to a larger subset of $X^P$, encouraging stronger generalization effects.

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 are relatively simple?

Formally, we denote a function $\Omega_{\text{regional}} : \Theta \to \mathbb{R}$ defined as,

$$\Omega_{\text{regional}}(\theta) \triangleq \frac{1}{R} \sum_{r=1}^{R} \omega_r \cdot \Omega_r(\theta)$$

where $\Omega_r : \Theta \to \mathbb{R}$ can be written as,

$$\Omega_r(\theta) \triangleq \text{PathLength}(X_r, f(\cdot), h_r).$$

$R$ is number of regions that partition $X^P$. $h_r$ is a hyperparameter for the decision tree in region $r$, and each $\omega_r \in \mathbb{R}$ 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, $\Omega_{\text{regional}}$ is equivalent to $\Omega_{\text{global}}$ on the subset $X_r$ as opposed to the full dataset $D$.

Referring to Alg. 1 for computing APL, we define $\text{TrainTree}$ as the DecisionTreeClassifier module distributed in Python’s Scikit-Learn with a fixed random seed. Notably, $\text{TrainTree}$ 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,

$$h_r = \min\left\{ \frac{N_r}{C} + 1 \right\}$$

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 $\theta$, 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 $\Omega_{r}^{\text{regional}} : \Theta \to \mathbb{R}^+$ that maps a parameter vector $\theta \in \Theta$ to an estimate of $\Omega_r(\cdot)$, the APL in region $r$. In our case, $\Omega_r(\cdot)$ is implemented as a shallow multi-layer perceptron, and is thus differentiable:

$$\Omega_r(\theta, \phi_r) \triangleq \text{MLP}_r(\theta; \phi_r)$$

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,

$$\Omega_{\text{local}}(\theta; \phi_1, \ldots, \phi_R) \triangleq \frac{1}{R} \sum_{r=1}^{R} \text{MLP}_r(\theta; \phi_r)$$

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

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

for all $r = 1, \ldots, R$ where $\theta_j$ is sampled from a dataset of $J$ known parameter vectors and their true APLs: $\Omega_r^{\text{regional}}(\theta_j)$. This dataset can be assembled using the candidate $\theta$ 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, $(\text{MLP}_r)_{r=1}^R$, 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
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 $\Omega^\text{regional} \cdot (\cdot)$ 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 surrogate (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 $D^\theta$ of the most recent $\theta$ is insufficient to learn the mapping from a thousand-dimensional weight vector to the APL. Using stale (very old) $\theta$, 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 $\theta$, we sample from a Dirichlet distribution with $J$ categories and form a new parameter as a convex combination of the elements in $D^\theta$. 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 $\text{TrainTree}$. 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.

Figure 2: (a-d) Decision trees using randomized training; (e-h) decision trees using deterministic 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.
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 θ ∈ Θ, Ω^regional(θ) and Ω^global(θ) 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 Ω^regional(θ) as an evaluation criterion. In this context, Ω^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, Ω^regional(θ).

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 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 upwards. The training dataset is relatively small for an over-parameterized neural network. Second, we find that L2 is not a desirable regularizer: we use \( \lambda \) means clustering model with \( \lambda \) regional tree regularization, it comes with a large increase in accuracy.

### 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).

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Acc.</th>
<th>Test APL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unregularized</td>
<td>0.8296</td>
<td>17.9490</td>
</tr>
<tr>
<td>L2 (( \lambda = 0.001 ))</td>
<td>0.8550</td>
<td>16.1130</td>
</tr>
<tr>
<td>Global Tree (( \lambda = 1 ))</td>
<td>0.8454</td>
<td>6.3398</td>
</tr>
<tr>
<td>Regional Tree (( \lambda = 0.1 ))</td>
<td>0.9168</td>
<td>10.1223</td>
</tr>
</tbody>
</table>

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 \( D \) to fit a k-means clustering model with \( k = 5 \). Each example \( x_n \in D \) is then assigned a number, \( s_n \in \{1, 2, 3, 4, 5\} \). We define \( X_s = \{x_n|s_n = r\} \subseteq \mathcal{X}^P \).

(1) **Bank Marketing** (Bank): 45,211 rows collected from marketing campaigns for a bank [22]. \( x_4 \) 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 [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 \( \lambda \) 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 \( D \) 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 that...
We study 11,786 intensive care unit (ICU) patients from the MIMIC III dataset [9]. We ignore the temporal dimension, resulting in a dataset \( \mathcal{D} = \{x_n, y_n\}_{n=1}^N \) with \( P = 35 \) input features, and \( Q = 4 \) binary outcomes. \( x_n \) measures continuous features such as respiration rate (RR), blood oxygen levels (paO\(_2\)), fluid levels, and more. \( y_n \) detail if vassopressin, sedation, mechanical ventilation, or renal replacement therapy were 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 above the mean, one standard deviation from the mean, and more than one standard deviation below 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 accuracy in all datasets. We find that in the lower APL area, 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.

## 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 \( \mathcal{D} = \{x_n, y_n\}_{n=1}^N \) with \( P = 35 \) input features, and \( Q = 4 \) binary outcomes. \( x_n \) measures continuous features such as respiration rate (RR), blood oxygen levels (paO\(_2\)), fluid levels, and more. \( y_n \) detail if vassopressin, sedation, mechanical ventilation, or renal replacement therapy were applied, respectively. Models are trained to 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.
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.

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 them 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 benefit from certain medications. 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 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 does.

7.2 HIV

We study 53,236 patients with HIV from the EuResist Integrated Database [30]. Each input \( x_n \) contains 40 features, including blood counts, viral load measurements, and lab results. Each output \( y_n \) 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/mm³, 200 - 300 cells/mm³, 300 - 500 cells/mm³ and >500 cells/mm³ [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 medium 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 too restrictive, resulting in no benefit from using a deep model. Across all experiments, we see that region regularization is more adept at finding medium 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 \( x_n \in D \) has its own region i.e. local explainability.
Regional Tree Regularization for Interpretability in Black Box Models

Regularized deep models outperform decision trees. If we compare

<table>
<thead>
<tr>
<th>Bank</th>
<th>Gamma</th>
<th>Adult</th>
<th>Wine</th>
<th>Crit. Care</th>
<th>HIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fidelity</td>
<td>0.892</td>
<td>0.881</td>
<td>0.910</td>
<td>0.876</td>
<td>0.900</td>
</tr>
</tbody>
</table>

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 \( \mathcal{D}_o \), (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 \( \mathcal{D}_o \), 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 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


