Multimodal Prescriptive Deep Learning

Anonymous authors
Paper under double-blind review

Abstract

We introduce a multimodal deep learning framework, Prescriptive Neural Networks (PNNs), that combines ideas from optimization and machine learning, and is, to the best of our knowledge, the first prescriptive method to handle multimodal data. The PNN is a feedforward neural network trained on embeddings to output an outcome-optimizing prescription. In two real-world multimodal datasets, we demonstrate that PNNs prescribe treatments that are able to significantly improve estimated outcomes in transcatheter aortic valve replacement (TAVR) procedures by reducing estimated postoperative complication rates by 32% and in liver trauma injuries by reducing estimated mortality rates by over 40%. In four real-world, unimodal tabular datasets, we demonstrate that PNNs outperform or perform comparably to other well-known, state-of-the-art prescriptive models; importantly, on tabular datasets, we also recover interpretability through knowledge distillation, fitting interpretable Optimal Classification Tree models onto the PNN prescriptions as classification targets, which is critical for many real-world applications. Finally, we demonstrate that our multimodal PNN models achieve stability across randomized data splits comparable to other prescriptive methods and produce realistic prescriptions across the different datasets.

1 Introduction

Today's society provides an increasing availability of large quantities of data, particularly multimodal data consisting of structured and unstructured elements. As a result, developing systematic and personalized decision-making methods that can leverage such multimodal data becomes more and more critical, and the benefits of data-driven methods become more and more visible. For example, medical professionals could systematically and optimally treat patients based on individual characteristics, clinical notes, and medical scans (Soenksen et al., 2022). Companies in technology and digital advertising would be able to increase customer impact by customizing content and advertisements according to user-specific data. In the retail industry, such personalized models would allow companies to dynamically price goods and services based on the user or environment for increased revenue.

Much of the current work in machine learning and deep learning focuses on improving the accuracy of output prediction. We find that deep learning has tremendous and underutilized potential in the area of decision-making. This paper combines ideas from machine learning and optimization to move from prediction to prescription, with the ability to leverage multimodal data. We introduce a novel, multimodal, deep learning framework that we call a Prescriptive Neural Network (PNN). In this paper, we demonstrate how our models handle complex data structures and how effective they are in both multimodal and unimodal real-world applications. Through these applications, we show that our PNN models are flexible with different treatment scenarios that cover all real-life application settings. Our models also give stable and realistic results, comparable to existing prescriptive methods, and provide the user with more control over the resulting prescriptions. On tabular datasets in particular, we are able to recover interpretability by applying knowledge distillation and fitting interpretable Optimal Classification Tree models (Bertsimas & Dunn, 2017; 2019) on the PNNs' prescriptions as classification targets; we find that these Mirrored OCTs perform comparably to their PNN counterparts, meaning that interpretability comes with little cost to performance.

1.1 Related Literature

Previous literature in data-driven personalized decision-making includes the Regress & Compare framework, tree-based methods, and causal methods.

Regress & Compare. Regress & Compare is a black-box methodology where a regression model is trained to predict the outcome under each treatment. The set of features used for training is the augmented feature data combined with the historical treatment given. Given an input observation and possible treatment options, the model is then used to select the treatment with the lowest (highest) outcome for a minimization (maximization) problem.

In this area, Zhao et al. (2012) introduce a framework that aims to estimate individual treatment rules. The goal is to assign treatments that maximize (or minimize) the expected outcome for each individual, by estimating the potential outcomes under each treatment (using an outcome weighted learning approach) and selecting the treatment that leads to the best outcome.

There are also many applications of the Regress & Compare methodology for prediction – examples include energy economics (Ferkingstad et al., 2011) and multidrug-resistant tuberculosis (Siddique et al., 2019). Other works use the Regress & Compare framework to move from predictions to prescriptions. In particular, Bertsimas & Kallus (2020) extends Regress & Compare solutions for prescriptive problems and incorporates k-nearest neighbors regression (kNN Altman (1992)), local linear regression (LOESS Cleveland & Devlin (1988)), classification and regression trees (CART Breiman et al. (1984)), and random forests (RF Breiman (2001)). Although Bertsimas & Kallus (2020) demonstrates that their methods are widely applicable and computationally tractable under mild conditions, we note that these are classical machine learning methods and do not take advantage of neural networks.

More specific applications of Regress & Compare for prescriptive problems include healthcare (Bertsimas et al., 2017a; Bayati et al., 2014) and revenue management (Bertsimas & Kallus, 2022). Bertsimas et al. (2017a) considers personalized diabetes treatment, while Bayati et al. (2014) combines prediction and decision-making to allocate interventions for post-discharge patients that were admitted due to heart failure. Bertsimas & Kallus (2022) considers the problem of optimal pricing, where they learn from historical observational data to optimize predicted revenue given price.

One possible limitation of the Regress & Compare approach is that it is affected by the number of samples per treatment, since for each treatment, only the samples that received that treatment in real life are considered. Also, it does not address the potential treatment assignment bias present in the data; e.g. healthier patients tend to receive lighter treatment and to have better outcomes. This is discussed in more detail in Section 2.3. Furthermore, the black-box nature reduces its interpretability, which is important for many real-world applications.

Tree-based methods. Kallus (2017) introduces Personalization Trees, which extend the Regress & Compare method for the problem of choosing the treatment with the best causal effect from a finite number of discrete options. Kallus presents three different recursive-partitioning-based algorithms: a greedy Personalization Tree, a Personalization Forest that bags Personalization Trees, and a globally optimal Personalization Tree. As these are tree methods, we note that they preserve interpretability.

Bertsimas et al. (2019) introduces Optimal Prescriptive Trees, which are similar to Personalization Trees but combine the counterfactual estimation and prescriptive learning tasks together in one training process and extend the framework of Optimal Classification Trees from Bertsimas & Dunn (2017; 2019). As a result, the trees are highly interpretable. Amram et al. (2022) further explores the optimal trees methodology and proposes Optimal Policy Trees, in which counterfactual estimation is performed separately from the prescriptive learning task. This allows for greater flexibility in discrete and continuous treatments, as well as better learning of the prescriptive task due to reduced complexity that results from the separation of the two training tasks. Like Optimal Prescriptive Trees, this method also preserves interpretability. These approaches, however, struggle with learning more complicated functional forms and are therefore limited to learning outcome functions that can be modeled by trees of relatively small depth.

Causal methods. This family of methods originates from the causal inference literature and includes both individual trees (causal trees) and their combinations (causal forests). Athey & Imbens (2016) introduces causal trees, which employ a recursive partitioning approach of the feature space to split the data into groups with similar treatment effects. Causal forests extend causal trees and represent a prescriptive black-box method that builds on random forests, as introduced by Wager & Athey (2018). While random forests are constructed from decision trees, causal forests are composed of causal trees, which aim to maximize the difference in outcomes between two treatments at each node during tree growth. The resulting outcomes are interpreted relatively to one another. In the binary treatment case, since there are only two options (treatment or no treatment), one option will yield a positive effect (outcome) and the other a negative effect. If the goal is to minimize the outcome, the treatment option with the negative effect is prescribed.

Other models in the causal inference literature include causal boosting (Powers et al., 2018) and causal MARS (Powers et al., 2018). However, the estimation of treatment effects, which is achieved by causal models, is not an explicit policy prescription, which is the goal of this work.

Finally, another approach by Zhou et al. (2023) takes inspiration from the causal inference literature and uses inverse propensity weight estimators to calculate the counterfactuals. Decision trees (both greedy and fully optimal) are then used for policy learning. Fully optimal trees, however, struggle with scalability, while the heuristic-based trees do not guarantee the best possible policy (optimality).

Deep learning methods. Other deep learning approaches to the optimal prescription problem include Patil et al. (2024); Sun & Tsiourvas (2023). Patil et al. (2024) introduces prescriptive networks that are shallow neural networks to address the binary treatment regime, in which a treatment may or may not be given. Their networks are optimized by overestimating conditional average treatment effects (CATE), and they propose a method using mixed-integer programming (MIP) to implement their networks into commercial solvers. Sun & Tsiourvas (2023) proposes a piecewise linear neural network model to output optimal prescriptions from a set of discrete treatments and show that their model partitions the input space into disjoint polyhedra, where all observations in the same partition are assigned the same treatment. Bergman et al. (2022) proposes a solver that takes as input user-specified pretrained predictive models (including neural networks) and formulates optimization models directly over those predictive models to provide final prescriptions.

We note that, like our PNN models, all of these works successfully combine ideas from optimization and machine learning. However, Bergman et al. (2022) does not incorporate Deep Learning in the prescriptive part of the framework, but only to generate predictions. Patil et al. (2024); Sun & Tsiourvas (2023) consider binary and discrete treatments respectively, while our work handles more treatment and outcome scenarios. Furthermore, our approach differs in the network's objective function used for training.

1.2 Contributions

Our contributions are as follows:

1. Combining machine learning and optimization, we propose a novel, multimodal, deep learning framework we call a Prescriptive Neural Network (PNN); to the best of our knowledge, our model is the first prescriptive method to handle multimodal data.

2. In two real-world multimodal datasets, we demonstrate that PNNs prescribe treatments that are able to significantly improve estimated outcomes in transcatheter aortic valve replacement (TAVR) procedures by reducing estimated postoperative complication rates by 32% and in liver trauma injuries by reducing estimated mortality rates by over 40%. Additionally, PNNs either outperform or perform comparably to existing, state-of-the-art, prescriptive methods on four real-world unimodal (tabular) datasets that span all four treatment scenarios: diabetes management (multiple continuous treatments), groceries pricing (single continuous treatment), splenic injuries treatment (multiple discrete treatments), and REBOAs in blunt trauma patients (binary treatment).

3. On tabular datasets we recover interpretability through knowledge distillation; we train Optimal Classification Trees (OCT) (Bertsimas & Dunn, 2017; 2019) on the feature data but using the PNN prescriptions as target classes, similar to a binary or multiclass classification task. We call these

Mirrored OCTs. Remarkably, the performance of the Mirrored OCTs is equally strong as that of the original PNNs, with a decrease in improvement of only 1.38% on average across the tabular datasets; this implies that interpretability may be recovered with minimal cost to performance.

4. Finally, we demonstrate that our multimodal PNN models achieve stability across randomized data splits comparable to other prescriptive methods and produce realistic prescriptions across the different datasets.

2 Methods

In this section, we review the methodology of our PNNs. We first formally define the prescriptive problem we seek to solve (Section 2.1), and then we present the training process, which is divided into four main steps: embedding extraction (Section 2.2), counterfactual estimation (Section 2.3), prescription policy learning (Section 2.4), and interpretability recovery (Section 2.5).

2.1 Problem definition

Formally, we consider a prescription problem, which can be characterized by observational data in the form $\{(\boldsymbol{x}_i, y_i, t_i)\}_{i=1}^n$:

- **Features** $\boldsymbol{x}_i \in \mathbb{R}^p$ is the p-dimensional feature data for the i-th observation.
- **Treatment** $t_i \in \mathcal{T}$ is the treatment applied historically to the i-th observation, where $\mathcal{T}$ is the set of all possible treatments. As treatments may be discrete or continuous, there are four possible treatment scenarios: binary (treatment or no treatment), multiple discrete (two or more treatment options), single continuous (one treatment option with continuous values), or multiple continuous (two or more treatment options, some or all taking on continuous values).
- **Outcome** $y_i \in \mathbb{R}$ is the result observed after treatment $t_i \in \mathcal{T}$ has been applied to the i-th observation.

Given this observational data, the aim is to develop an optimal prescriptive model that outputs a treatment $t \in \mathcal{T}$ that results in an optimal outcome y for each input observation $\boldsymbol{x}$.

2.2 Embedding Extraction

The first step in the model pipeline is to extract embeddings from the structured and unstructured data.

2.2.1 Structured data

We extract embeddings from structured feature data through traditional preprocessing techniques as described below, where the technique depends on whether the feature is numerical, categorical, or ordinal.

- **Numerical features.** Numerical features are normalized to the interval [0,1] by subtracting the minimum feature value and dividing by the feature range; we do this to increase stability and equal weighting of features during counterfactual estimation and model. We note that since tree models are independent of data scale, we use the original feature values when training all tree models, which ensure interpretability in the tree splits.
- **Categorical features.** For categorical features, we use one-hot encodings to convert them to binary features, such that each category becomes a new indicator feature.
- **Ordinal features.** Ordinal features are categorical features whose values carry numerical information. Since these categories have a natural order to them, we can assign each category a number such as 1 to 5, where relative magnitude holds information. The feature value assigned to the number "1" conveys that that value is less than that of a feature value assigned the number "4." These ordinal features are then treated as numerical features in our experiments.

2.2.2 Unstructured data

We extract embeddings from unstructured data using pretrained, deep learning models. By passing each observation's unstructured data through these pretrained models, we can obtain a vector representation of the unstructured datapoint. In particular, our experiments on medical data in Section 3 use Clinical Longformer (Li et al., 2022), a long sequence transformer model trained via a sparse attention mechanism on domain-specific, large-scale clinical corpora; from this model, we obtain a 768-dimensional embedding vector for each observation's text data.

We then leverage Principal Component Analysis (PCA) for dimensionality reduction purposes. Though optional, this step helps to improve stability, tractability, and performance in our PNN model, since the embeddings extracted from the pretrained models can be high-dimensional, which increases the complexity of the problem when the dataset size is comparable to the embedding dimension. In particular, we reduce to a 32-dimensional representation for the clinical notes in our experiments on medical data, but the size can be adjusted, depending on the application and the dataset size.

While we specifically use ClinicalLongformer, any pretrained large language model (LLM) may be used to process unstructured text data. Additionally, any pretrained computer vision (CV) model may be used to process unstructured image data. This results in an embedding extraction step for unstructured data that is not only highly accessible, but also highly flexible.

To get the final multimodal embeddings, we concatenate the individual modalities' embeddings to obtain one large embedding vector.

2.3 Counterfactual Estimation

The next step is counterfactual estimation. Because the prescriptive problem's dataset only contains historical observational data, the counterfactuals are unknown, e.g. the hypothetical outcomes $y(\boldsymbol{x}_i, t)$ for $t \neq t_i$, for each observation $\boldsymbol{x}_i$. We therefore perform a counterfactual estimation step (Dudik et al., 2011) that estimates the outcomes for each observation under every treatment. This produces a rewards matrix Γ, where $\Gamma_{i,t}$ is the estimated outcome of applying treatment t to the ith observation. The estimation process is slightly different for discrete and continuous treatments.

2.3.1 Counterfactual estimation of discrete treatments

We use two methods for counterfactual estimation of discrete treatments. The doubly-robust method is, however, preferred for almost all of the experiments in Section 3, since it addresses the treatment assignment bias. The two methods are as follows:

1. **Direct Method.** This method directly learns the outcome function $y_t(\boldsymbol{x})$ by training separate models, one for each treatment t. During training, each model uses only the subset of the observations that received treatment t. These models can be random forests or boosting methods and output an estimated outcome $\hat{y}_t(\boldsymbol{x})$ for when treatment t is hypothetically applied to observation $\boldsymbol{x}$.

2. **Doubly robust estimation.** Because direct estimation is often prone to treatment assignment bias, the doubly-robust estimator attempts to mitigate this bias by re-weighting the estimated direct outcomes with propensity score probabilities. This reweighting is expressed in Equation (1), which calculates the doubly-robust reward matrix Γ:

$$\Gamma_{i,t} = \hat{y}_{i,t} + \mathbb{1}\{t_i = t\}\frac{1}{p_{i,t}}(y_i - \hat{y}_{i,t}), \tag{1}$$

   where $\hat{y}_{i,t} = \hat{y}_t(\boldsymbol{x}_i)$ is the estimated outcome of sample i under treatment t, $p_{i,t} = \mathbb{P}[t_i = t]$ is the probability that treatment t is assigned to observation i in real life and y_i is the actual outcome of observation i. To reduce the potential instability that arises when we divide with the probability $p_{i,t}$, we clip $\frac{1}{p_{i,t}}$ that exceed a certain value (Lee et al., 2011).

For binary outcomes, we use classifiers for the counterfactual estimation, while for continuous outcomes, we use regressors.

2.3.2 Counterfactual estimation of continuous treatments

For continuous treatments, we train a regression model to predict the outcome of the ith observation using as input the observational data $\boldsymbol{x}_i$ and continuous prescribed treatment doses $T_{i,t}$ for each treatment t. Then, by discretizing the continuous dose values and only considering a subset of them as valid treatments, we use the trained model to retrieve the estimated outcome for the ith observation under all valid treatment schemes. This is analogous to real-world treatment scenarios; when we handle continuous treatments, we always select a subset of the possible ones, since the real-world treatment options need to be finite.

2.4 Prescription policy learning through feedforward neural networks

At its core, the architecture of our Prescriptive Neural Network (PNN) is the classical feedforward neural network trained via backpropagation (Rosenblatt (1958); Rumelhart et al. (1986)). Without loss of generality, we assume our goal is to minimize outcomes in the prescriptive problem. The objective of our prescriptive neural network is to minimize total rewards for the prescriptions $\tau(\boldsymbol{x}_i)$ assigned by the network to each observation $\boldsymbol{x}_i$ in the dataset:

$$\min_{\tau(.)} \sum_{i=1}^{n} \sum_{t \in \mathcal{T}} \mathbb{1}\{\tau(\boldsymbol{x}_i) = t\} \cdot \Gamma_{i,t}, \tag{2}$$

proposed by Amram et al. (2022). Because the indicator function is not differentiable, the backpropogation algorithm cannot handle Equation (2) exactly. We therefore "soften" the objective and leverage an approach analogous to that of multi-classification networks. The PNN assigns treatments probabilistically, such that its output layer consists of $|\mathcal{T}|$ neurons, one for each distinct treatment (as multi-classification networks have an output corresponding to each target class). We denote the output vector of the PNN as $\boldsymbol{z} \in \mathbb{R}^{|\mathcal{T}|}$ and apply a softmax activation function to these output neurons to obtain a probability distribution over the distinct treatments.

We obtain the final prescription of the network by finding the treatment t with the highest probability $\mathbb{P}[\tau(\boldsymbol{x}_i) = t] = \sigma_t(\boldsymbol{z})$. This approach is analogous to a classification network, where the predicted class is the one with the highest probability among the network's output nodes. The tractable objective for our PNN models is therefore:

$$\min_{\tau(.)} \frac{1}{n} \sum_{i=1}^{n} \sum_{t \in \mathcal{T}} \mathbb{P}[\tau(\boldsymbol{x}_i) = t] \cdot \Gamma_{i,t}. \tag{3}$$

2.4.1 Convergence properties

The loss function 3 shares key properties with the cross-entropy loss function commonly used in multiclass classification problems. The cross-entropy loss is defined as:

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}} y_{i,t} \sigma_t(\boldsymbol{z}), \tag{4}$$

where $y_{i,t}$ is the true label of sample i, and $\sigma_t(\boldsymbol{z})$ is the softmax probability for class t. The cross-entropy loss is widely used because it is smooth, has bounded gradients, and is Lipschitz continuous. These properties contribute to the efficient convergence of optimization algorithms like SGD and Adam (Kingma & Ba, 2014; Bottou, 2010).

Similarly, the loss function 3 employed in this work exhibits these desirable properties. Specifically:

- Smoothness: The loss function is smooth because it is a linear combination of the softmax probabilities $\sigma_t(\boldsymbol{z})$, which are themselves smooth functions (Bishop & Nasrabadi, 2006).

- Bounded Gradients: The gradients of the loss function are bounded, since the weights Γ_{it} are bounded due to clipping (as described in Section 2.3.1) and the derivative of the softmax function is also bounded (Goodfellow et al., 2016).
- Lipschitz Continuity: The loss function is Lipschitz continuous because the softmax function is Lipschitz continuous, and the weights Γ_{it} are bounded (Nesterov, 2013).

The primary difference between our loss function and the cross-entropy loss is that our loss function uses weights $\Gamma_{i,t}$ instead of true labels $y_{i,t}$, and it does not include the logarithm of the probabilities. However, these differences do not fundamentally alter the smoothness, boundedness, or Lipschitz continuity of the loss function. As a result, the convergence behavior of our loss function is similar to that of the cross-entropy loss when training feedforward neural networks for multiclass classification problems (LeCun et al., 2015).

Under the assumption of bounded weights in the network, a property typically observed when training with SGD or Adam (Ghadimi & Lan, 2013; Reddi et al., 2019), the network will converge to critical points of the loss function. This is consistent with the behavior observed in standard neural network training with cross-entropy loss (Zhang et al., 2016).

2.5 Recovering Interpretability with Optimal Classification Trees

For structured datasets, we are able to recover interpretability through the use of knowledge distillation, in which we fit Optimal Classification Trees (Bertsimas & Dunn, 2017; 2019) on the feature data and prescription outputs of the PNN. We present an example of such a Mirrored OCT in Figure 1 (with other examples available in Appendix A.4). This example comes from the REBOA (resuscitative endovascular balloon occlusion of the aorta) application in Section 3.5. In this real-world problem, we aim to minimize patient mortality by either prescribing (treatment 1) or not prescribing (treatment 0) the REBOA treatment. The tree in the figure is fit on the same observational data used to train its corresponding PNN, while the PNN prescriptions are used as target classes. We observe that this tree is very interpretable, and the features chosen for the splits come from our structured, observational data. If a patient (sample) is assigned to a leaf where the prediction is 0, then they are not prescribed the treatment, and if they are assigned to a leaf where the prediction is 1, the REBOA treatment is recommended.

3 Experiments with real-world datasets

In this section, we apply PNNs on real-world datasets that are both multimodal and unimodal. We first review methodology for data splits, network architecture, and performance evaluation, which are relevant for all of our experiments. We then report results and relevant discussions for each of our two multimodal datasets and four unimodal datasets.

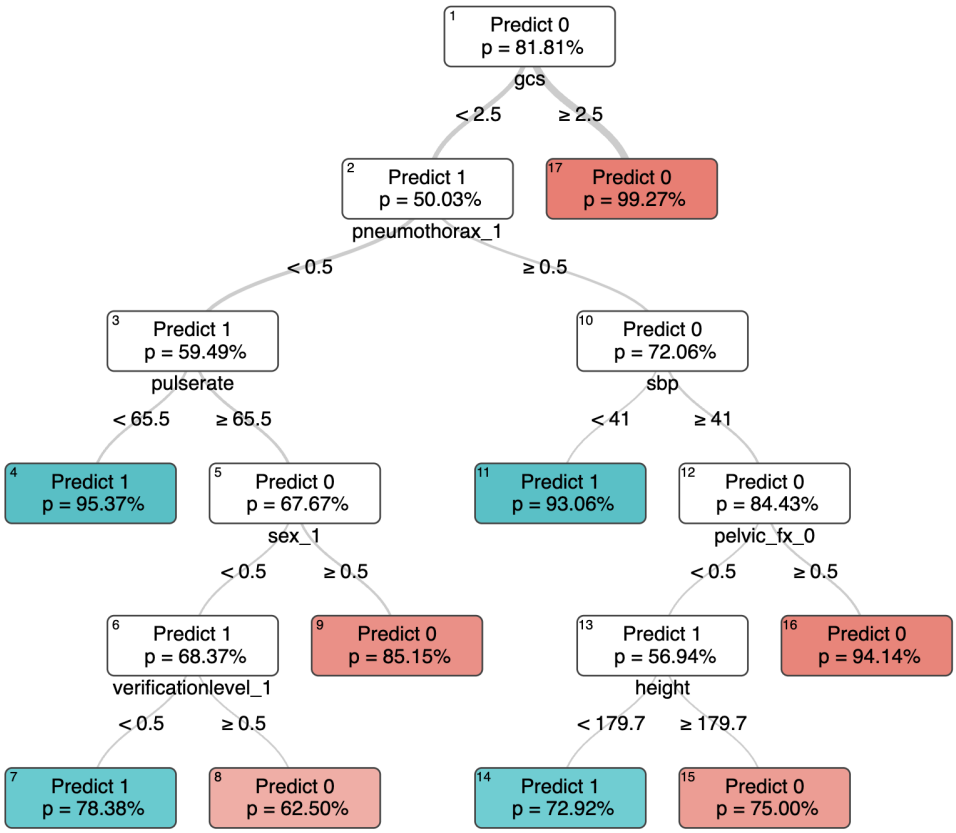
3.1 Data splits

We split each dataset evenly, using 50% for the training and 50% for the test set. This choice is attributed to the fact that in order to evaluate the performance of the prescriptive methods on a test set, knowing the outcomes of the samples under the different treatments is required. Furthermore, to prevent data leakage between training and test splits and unfair performance, the counterfactual estimation for the test set is necessary and performed separately from that of the training set. Because we need enough data points to ensure high-quality counterfactual estimation of both the training and test sets, the typical ratios of 80-20 or 70-30 are not applicable in this case.

3.2 Network Architecture

We uniquely tune the PNN architecture for each dataset. We specify and finetune the hyperparameters of the PNN in Appendix A.1. In addition to these architectural choices, we choose the Adam optimizer. We tune the aforementioned hyperparameters using a validation set we extract from the training set that is not

Figure 1: Example of REBOA Mirrored OCT.



used in model training. We typically keep 15% of the training data for the validation set. The Mirrored OCTs are then trained on the prescriptions from each of the PNN models.

3.3 Experiments

To evaluate the models' performance on each dataset, we perform multiple train-validation-test splits and report the average performance of each split's models. This ensures that the results are not tailored to a specific data split, and also enables the investigation of stability of the prescriptive methods. Also, given the randomness often associated with training machine learning models, we train multiple models per split and also average their performance. In total, we perform 5 randomized data splits per dataset, and we train 5 models per split, so we train 25 models in total, for each model type.

3.4 Performance metrics

To assess model performance, we use a relative outcome improvement metric measured on the unseen test set. This metric compares the estimated outcome of the treatment prescribed by the model with that of the real-life treatment. For most datasets, where doubly robust estimation is applied, test-set reward matrix entries lack natural meaning. Thus, instead of comparing with the actual outcome and to ensure fairness, hypothetical outcomes for both treatments (model-prescribed and real-life) are drawn from the reward matrix and then compared. The average relative outcome improvement is then computed as:

$$\bar{I} = \frac{\sum_{i=1}^{n} |\Gamma_{i,\hat{t}_i} - \Gamma_{i,t_i}|}{\sum_{i=1}^{n} \Gamma_{i,t_i}}, \tag{5}$$

where $\Gamma_{i,\hat{t}_i}$ is the estimated outcome for the ith observation if the prescribed treatment $\hat{t}_i \in \mathcal{T}$ is applied and Γ_{i,t_i} is the estimated outcome for the ith observation under treatment t_i.

For the case of unstructured datasets, the evaluation on the test set can be performed using counterfactuals that have been calculated either by using a single modality, or multimodal data. To address that, the outcome improvement is evaluated using the test set counterfactuals with both types of models (in this case, tabular, and multimodal, from tabular and notes).

3.5 Unstructured datasets

We demonstrate the efficacy of our PNN models on two real-world, multimodal datasets: transcatheter aortic valve replcement (TAVR) and liver trauma injuries. Both datasets include tabular (structured) and clinical notes (unstructured) data, and we train two models: unimodal models – fit on just the tabular data – and multimodal models – fit on the combined tabular and notes data. We report the results of both datasets and discuss the improved performance of the multimodal pipeline in this section. To ensure that there is no bias stemming from which modality is used to perform the reward estimation in the test set, we report the estimated outcome improvement under rewards estimated using both the single modality (tabular data) and the multiple modalities (tabular and notes).

Transcatheter aortic valve replacement (TAVR). Transcatheter aortic valve replacement (TAVR) is a treatment option for patients with severe aortic stenosis across all levels of surgical risk. In the United States, two transcatheter heart valves (THV) are used, the balloon-expandable Edwards Sapien 3 and the self-expanding Medtronic Evolut Pro Plus. Selection of valve choice by medical professionals is generally based on several factors including operator preference, patient characteristics, and valvular/annular anatomy on a computerized tomography (CT) scan (Mitsis et al., 2022; Leone et al., 2023). Despite improvement in TAVR devices, implant techniques, and operator experience, permanent pacemaker implantation (PPI) continues to remain a frequent complication with an estimated prevalence of 7-18% (Webb & Wood, 2012; Smith et al., 2011), with potential consequences on patients' mortality and cost of care. This dataset contains demographic (e.g. age, sex, bmi) and medical information (e.g. hypertension, Left Ventricular Ejection Fraction), as well as radiology reports of echocardiograms and CT scans from 2,127 patients, and the problem we seek to solve is prescribe the most appropriate type of valve to patients, so that their risk of PPI is minimized. We train two different sets of models, one where only the tabular features are considered, and one where notes are also incorporated, in the form of embeddings, extracted as described in Section 2.2.2.

Liver trauma injuries. Acute liver injury is considered one of the two most common solid organ injuries in blunt trauma victims. However, inaccuracies exist in the grading of liver injuries by human read and interpretation of CT scans, which may lead to mistreatment (Georg et al., 2014). Therefore, personalized treatment for the patient is important in trauma management. This dataset comes from electronic medical records of 722 liver injury patients and includes features such as patient demographics, history of illness, lab results, and allergies. We aim to prescribe either surgical or non-surgical intervention to minimize patient mortality (binary outcome).

The results for both datasets are presented in Table 1. For both of the datasets, we observe that the multimodal models significantly outperform the unimodal ones, by even more than 10% on average, demonstrating the benefit of increased information from the added language modality. Although the discrepancy between the outcome improvement can be quite different depending on the modality used to train the test set reward estimators, there is a clear improvement observed when multimodality is employed under both estimators. The results are also stable, across 5 different data splits and 5 different models per split, and indicate the prescriptive power multimodality can offer.

Mirrored OCTs, trained on the PNNs' prescriptions, result in an even larger outcome improvement compared to the PNNs, in both datasets. This is a dataset-specific observation, that demonstrates, however, that Mirrored OCTs do not generally result in performance decrease. We also quantify the approximation error between the PNN and OCT prediction using what we call the "training accuracy" of the OCTs on the prescriptions, which we present in Table 2. For example, a training accuracy of 79% means that in the

Table 1: Improvement(%) in risk for the experiments with unstructured data (TAVR, liver trauma), where lower risk is better. We report the average improvement and standard deviation across the five splits.

		TAVR models		**Liver trauma models**	
Estimator	**Method**	**Tabular**	**Multimodal**	**Tabular**	**Multimodal**
Tabular	PNN	3.41 ± 5.06	8.66 ± 6.80	16.34 ± 10.09	32.50 ± 12.40
	Mirrored OCT	4.84 ± 6.24	10.10 ± 9.02	24.86 ± 9.94	34.96 ± 6.80
Tabular & Notes	PNN	20.50 ± 3.86	32.15 ± 7.52	25.32 ± 5.10	40.30 ± 14.70
	Mirrored OCT	$\mathbf{21.33 \pm 10.43}$	$\mathbf{34.50 \pm 9.79}$	$\mathbf{37.46 \pm 5.87}$	$\mathbf{45.11 \pm 4.18}$

Table 2: Training accuracy (%) of the Mirrored OCTs for the unstructured datasets. We report the average accuracy and standard deviation across the five splits.

Dataset	**Tabular model**	**Multimodal model**
TAVR	79.03 ± 3.09	75.73 ± 0.69
Liver trauma	87.66 ± 3.50	84.07 ± 3.11

training data split, the OCT correctly classified 79% of the PNN's prescriptions. We observe that the OCT training accuracy is higher in the tabular compared to the tabular and notes model. This is to be expected, since the tabular model is trained using fewer features and thus is easier to be mirrored by a classification model.

3.6 Structured datasets

We now apply our PNN models to four real-world, unimodal tabular datasets: diabetes management, groceries pricing, splenic injuries treatment, and REBOA in blunt trauma patients. Because these are purely tabular datasets, we are able to recover interpretability by fitting Mirrored OCT models. For more details on the treatment scenarios covered by these four datasets, the counterfactual estimation method employed, as well as the training accuracy of the Mirrored OCTs (defined in the previous section), please refer to Table 10 in Appendix A.2.

Table 3: Improvement(%) for structured datasets. We report the average improvement and standard deviation across the five splits.

Method	**Diabetes**	**Groceries**	**Spleen**	**REBOA**
Regress & Compare	2.00 ± 0.25	94.17 ± 6.25	8.03 ± 2.24	-17.80 ± 27.91
Causal Forest	2.50 ± 0.69	98.68 ± 5.98	-4.84 ± 13.14	-4.82 ± 3.18
Optimal Policy Tree	1.72 ± 1.53	106.58 ± 5.33	16.41 ± 3.15	$\mathbf{10.79 \pm 4.34}$
PNN	$\mathbf{4.01 \pm 0.90}$	$\mathbf{110.88 \pm 5.91}$	$\mathbf{20.47 \pm 1.68}$	10.22 ± 4.55
Mirrored OCT	3.93 ± 1.21	110.22 ± 6.94	15.72 ± 6.01	10.21 ± 5.03

We present results for all four structured datasets in Table 3, where we directly compare PNNs and their Mirrored OCTs with the performance of other well-known, state-of-the-art prescriptive methods, including Optimal Policy Trees, Regress & Compare, and Causal Forests. We note that for Regress & Compare, we typically train an XGBoost Regressor or Classifier (depending on the nature of the outcome). For this purpose, we append the actual treatment as a separate column in the observational data and we train the predictive model to predict the real-life outcome under the treatment. To select the best treatment for a new sample, we append each of the available treatments separately and we obtain the final outcome in each case. The treatment that results in the best outcome is selected.

Diabetes management. This dataset is based on electronic medical records of 58,200 patients with type 2 diabetes from 1999 to 2014 from the Boston Medical Center. It contains information regarding patient demographics, a timeseries of insulin levels, as well as current drug prescriptions. Patient treatments include combinations of insulin, metformin, and oral blood glucose regulation agents, and patient outputs are the resulting hemoglobin A1C measurements (continuous outcome), for which lower values are more optimal.

Groceries pricing. For this study, we select the publicly-available retail dataset "The Complete Journey" (Lugauer et al., 2020; Biggs et al., 2021), which contains household-level transactions of many products over two years of 2,500 frequent-shopper households. We focus on one specific product, strawberries. The task here is to, given household demographics, prescribe optimal prices to strawberries with a binary outcome indicating if the household purchases strawberries or not after being assigned the strawberry price. The objective is to maximize revenue, where revenue is defined as the price if strawberries are purchased and zero otherwise (e.g., binary). After filtering the data to only the relevant households that had purchased strawberries at least once, the final dataset consists of 97,295 transactions. We impute strawberry prices for cases where strawberry-purchasing households did not purchase strawberries on that specific trip by using the mode of the strawberry prices on the most recent day prior to the trip on which no strawberries were purchased. We consider prices from \$2 to \$5, inclusive, in increments of \$0.50. Since there does not seem to be a strong correlation between strawberry price and the covariate features, rewards are estimated using the direct method.

Splenic injuries treatment. The spleen is an immunologic intra-abdominal organ on the left side of the body, which may be removed in the case of injury. In the 1970's to 1980's, the medical community saw a shift towards preservation of the spleen rather than removal, thus making it important to correctly determine if spleen removal was indeed necessary. This specific dataset includes data on spleen surgical operations, in addition to demographic and medical data consisting of numerical, binary, and categorical types. After preprocessing, we have 35,954 rows of patient data in this dataset. We aim to optimally prescribe splenectomy, angioembolization, or observation in blunt splenic injuries to minimize patient mortality (binary outcome).

REBOA in blunt trauma patients. The use of resuscitative endovascular balloon occlusion of the aorta (REBOA) for control of noncompressible torso hemorrhage continues to be highly debated. Being able to appropriately determine if such a treatment should be used is critical in order to decrease the misuse of the treatment in hemodynamically unstable blunt trauma patients. This dataset includes 9,998 patients, with features that are both demographic and medical in nature, including numerical, binary, and categorical values. The goal is to prescribe the REBOA treatment or not to minimize patient mortality (binary outcome). Some feature columns contain unknown values; we therefore use Optimal Imputation (Bertsimas et al., 2017b) with K-Nearest Neighbors to fill the missing values. A few features are integral, and we round imputed values to the nearest integer to maintain integrality.

As demonstrated in Table 3[1], we generally see an improvement in estimated outcome across all methods for all four structured datasets. Notably, we observe that PNNs and the Mirrored OCTs perform very strongly in all experiments. In particular, the PNNs outperform the other models in the diabetes, groceries and spleen datasets, while performing comparably to Optimal Policy Trees in the REBOA dataset. In especially the diabetes dataset, PNNs remarkably outperform other models by roughly twice as much; since the diabetes dataset falls under the multiple continuous treatment case, the exceptional PNN performance reflects its ability to handle more complicated data. We also note that the Mirrored OCTs also generally perform very similarly to PNNs, with the added benefit of interpretability.

4 Discussion

In general, reporting improvement based on estimated rewards is a good approximation for evaluating the performance of prescriptive methods. However, such metrics do not provide any insights into how realistic the prescriptions are or how different they are from policies observed in the data. Another critical aspect is

[1]For the groceries dataset, improvement is computed as mean revenue improvement rather than outcome improvement, where mean revenue is $\bar{p}_r = \frac{1}{n}\sum_{i=1}^{n} \Gamma_{i,\hat{t}_i} \cdot \hat{t}_i$, and actual revenue is $\bar{p}_r = \frac{1}{n}\sum_{i=1}^{n} y_{i,t_i} \cdot t_i$, $\hat{t}_i$ is the prescribed treatment for sample i by the model, and t_i is the real-life prescribed treatment.

the models' stability. The prescriptive models should be robust in their prescriptions across dataset splits and with respect to inherent randomness during training. Furthermore, interpretability is crucial for real-world deployment, as users of models must understand where decisions are coming from in order to implement them. Finally, we discuss briefly the potential and challenges for real-world deployment of PNNs and Mirrored OCTs. We therefore discuss these four topics – realistic and stable prescriptions, interpretability, as well as real-world deployment – in the following sections.

4.1 Realistic Nature of Prescriptions

Providing realistic prescriptions is crucial, particularly when employing prescriptive tools in practice. To evaluate the realism of the provided prescriptions, we quantify the deviation between the prescribed and real-life treatments of individual samples. This evaluation is carried out per model, by calculating the mean absolute difference between each sample's prescribed and real-life treatment throughout the dataset (training, validation, and test sets) and then averaging it across all samples. For discrete cases (REBOA and spleen datasets), the treatments are ordered in terms of severity, so that the distance makes sense as a metric. For this purpose, the $N_m = 25$ trained models from each dataset are considered. The mean absolute difference for the k-th individual model is given by:

$$D_k = \frac{1}{n} \sum_{i=1}^{n} \|\hat{t}_i - t_i\|_1, \tag{6}$$

where n is the size of the dataset, $\hat{t}_i$ is the prescribed treatment for sample i, and t_i is the treatment sample i got in real life. The mean absolute difference across the $N_m = 25$ models is then computed as:

$$\bar{D} = \frac{1}{N_m} \sum_{k=1}^{N_m} D_k. \tag{7}$$

The results are presented in Tables 4 and 5. Clearly, we prefer both high performance $\bar{I}$ and low mean absolute difference $\bar{D}$, as this ensures that improvements in outcomes are not achieved through disproportionate shifts in treatment assignments. We naturally expect that PNN-prescribed treatments are more different than those in real-life, as compared to the other prescriptive methods, since as presented in Sections 3.5 and 3.6, PNNs outperform the other methods in most datasets. Contrary to our expectation, however, we favorably observe that PNNs, as well as the Mirrored OCTs, result in mean absolute difference between the prescribed and the actual treatments that is comparable to the rest of the methods.

Table 4: Mean Absolute Difference between prescribed and actual treatments for structured datasets.

Method	**Diabetes**	**Groceries**	**Spleen**	**REBOA**
Regress & Compare	**0.4556**	1.033	**0.3355**	0.2815
Causal Forest	0.5923	**1.0042**	0.7761	0.2492
Optimal Policy Tree	0.7490	1.0221	0.3755	**0.1199**
PNN	0.4903	1.4386	0.5185	0.1568
Mirrored OCT	0.4936	1.4391	0.5069	0.1435

Table 5: Mean Absolute Difference between prescribed and actual treatments for unstructured datasets.

	TAVR models		**Liver trauma models**	
Method	**Tabular**	**Multimodal**	**Tabular**	**Multimodal**
PNN	0.4561	**0.6144**	0.3590	0.3837
Mirrored OCT	**0.4280**	0.6539	**0.3351**	**0.3089**

For unstructured data, Section 3.5 highlights the significant advantage of the multimodal approach over tabular-only methods. As expected, this performance edge indicates that the proportion of prescription changes is higher in the multimodal case. The results indicate that, in the TAVR case for example, around 62% of the PNN prescriptions change from one valve to the other in the multimodal case, and 45% in the tabular case, which is a considerable shift. For the liver trauma dataset, the difference between the prescribed and the actual treatments is smaller.

A critical advantage of neural networks is that the user has some control over how much the prescriptions change. For example, depending on the application, a threshold can be selected that limits the number of treatment assignment modifications, and only models that satisfy this constraint on the validation set are considered. Alternatively, one can incorporate a penalty term in the objective function, to penalize an excessive number of treatment switches. Alternatively, one can add different penalty terms in the loss function that account for different treatment constraints. This is application-specific, but highlights the flexibility that neural networks offer compared to other prescriptive methods.

The realism of prescriptions is also evaluated by examining the average number of distinct prescriptions per type of model, which shows how much the model is capable of utilizing the full treatment space. This is evaluated as:

$$\bar{N} = \frac{1}{|\mathcal{T}|} \cdot \frac{1}{N_m} \sum_{i=1}^{N_m} |t : \exists j : \hat{t}_j = t,\ j = 1, \ldots, n|, \tag{8}$$

where the average number of prescriptions is normalized by the size of treatment space, to calculate a percentage and thus make the metric comparable across the different datasets. The results are presented in Tables 6 and 7.

Table 6: Percentage of different prescriptions selected by the models for structured datasets.

Method	**Diabetes**	**Groceries**	**Spleen**	**REBOA**
Regress & Compare	36.92	23.33	66.67	70.0
Causal Forest	99.69	95.33	88.00	100.0
Optimal Policy Tree	59.38	48.67	85.33	100.0
PNN	16.62	49.33	81.33	100.0

Table 7: Percentage of different prescriptions selected by the models for unstructured datasets.

	TAVR models		**Liver trauma models**	
Method	**Tabular**	**Multimodal**	**Tabular**	**Multimodal**
PNN	100.0	100.0	100.0	100.0
Mirrored OCT	100.0	98.0	86.0	76.0

In most of the structured datasets, we observe that PNNs and Mirrored OCTs prescribe a high percentage of the available treatments, with the exception of the diabetes dataset, where the treatment options are multiple. The Regress & Compare approach clearly underutilizes the treatment space, since it prescribes a small percentage of treatments in most cases; this reveals potential treatment assignment bias that may not be mitigated through this approach. Causal Forests seem to make prescriptions that mostly cover the full treatment regime; however their performance is worse than Optimal Policy Trees and PNNs, as discussed in Section 3.6. In the unstructured case, both tabular and multimodal PNN models employ all of the available treatments. We observe that some of the Mirrored OCTs only prescribed one treatment in the liver dataset, but most of them prescribed both. Naturally, as the multimodal datasets are more complex, the resulting networks are harder to be explained by trees.

Most importantly, PNNs are flexible in this feature too; by using the dropout mechanism (Srivastava et al., 2014) in the last layers of the network, which is often used to prevent overfitting in neural networks, all of the output nodes are forced to be activated during training. As a result, more areas of the network are used, which empirically shows an increase in selected treatments by the model. The flexibility of PNNs is also underlined by the fact that they provide, for each observation, a probability of each treatment, similarly to a classification problem, where neural networks provide a probability of each class. The results presented for PNNs consider the prescription with the highest probability for each sample. However, one can employ a treatment-specific probability threshold to select the final treatment, like in classification problem; for example, this may be done according to some predefined, meaningful, treatment allocation percentage. This provides the user with some control over the resulting treatment distribution.

Overall, PNNs achieve a balance between performance and realism in prescriptions, and they also reasonably cover the treatment space. These are important factors that make them reliable for practitioners and their leverage in different real-life applications.

4.2 Stability

Given the randomness that is present when training neural networks, their stability compared to other machine learning models is often criticized (Colbrook et al., 2022). The goal of this section is to compare the stability between the different prescriptive approaches by measuring the standard deviation of each observation's treatment distribution, which results from the $N_m = 25$ different models that have been trained for each dataset. Ideally, the prescriptions should be consistent across the different model runs and data splits; otherwise the method is very sensitive to the training data distribution, which reduces the credibility of the prescriptions.

For each observation, the standard deviation of its prescriptions across the different N_m models is calculated, and we present averages across the different observations in Tables 8[2] and 9.

Table 8: Standard deviation of each sample's prescriptions distribution across $N_m = 25$ models for structured datasets.

Method	Diabetes	Groceries	Spleen	REBOA
Regress & Compare	**0.0752**	**0.0285**	**0.0940**	0.4230
Causal Forest	0.5139	0.1667	0.4535	0.3838
Optimal Policy Tree	0.5911	0.7227	0.1877	**0.0225**
PNN	0.2643	0.9332	0.3178	0.0679

Table 9: Standard deviation of each sample's prescriptions distribution across $N_m = 25$ models for unstructured datasets.

	TAVR models		Liver trauma models	
Method	Tabular	Multimodal	Tabular	Multimodal
PNN	**0.4062**	**0.4239**	0.3336	0.3202
Mirrored OCT	0.4563	0.4387	**0.2190**	**0.1032**

We observe that the standard deviation of PNNs' prescriptions is comparable to the other models across all datasets, which indicates that although training neural networks is associated with inherently more randomness (random weight initialization, stochastic gradient descent), they result in relatively consistent prescriptions across different data splits and different runs, similarly to the more deterministic prescriptive methods.

[2]For multiple continuous treatments (diabetes dataset), to get the standard deviation for each sample, we first calculate the standard deviation for each drug separately and then we average across the three drugs. For Regress & Compare, since we use XGBoost models, there is no randomness in each split, so the 5 models produce the same prescriptions. This explains why in most of the datasets, Regress & Compare has the lowest standard deviation.

In particular, for the unstructured datasets in Table 9, the standard deviation of Mirrored OCTs is considerably smaller than the PNNs' in the liver trauma dataset. We attribute this to the fact that Mirrored OCTs were likely not able to fully capture the PNNs' complexity and are therefore simpler in their decision-making rules. For the TAVR dataset, the deviation is consistent across the different modalities, which indicates that adding a modality does not impact the model's stability.

4.3 Interpretability

We discuss now the interpretability of PNNs that is recovered via knowledge distillation of the Mirrored OCTs. In particular, we discuss the unstructured TAVR dataset, for which we can partially recover interpretability, and for the structured diabetes management dataset. Please refer to Appendix A.3 for similar analyses for our other datasets and Appendix A.4 for model visualizations.

TAVR. We discuss the multimodal Mirrored OCT from Figure 2 for the TAVR dataset. While the embedding features from the clinical notes are not interpretable, we can still recover some interpretability through the tabular features selected by the Mirrored OCT. We see that the OCT selects one note feature, as well as the Valve-to-Annular Aortic Valve Area ratio (VDAoVA), age, and difference between the annular area of the patient's native aortic valve and the area of the prosthetic valve being implanted (Area Oversize). We observe that the OCT provides an insight into the prescriptions made by the PNN, even when it is trained on the multimodal data.

Diabetes management. We consider and compare the Mirrored OCT from Figure 4 and the Optimal Policy Tree from Figure 5 for diabetes management. Both models were trained with the same data, methods, and parameters as in Section 3.6. For visualization reasons, Figure 4 displays a portion of the tree. From Figure 4, we can see that the features selected by the tree include "HbA1c_mean" (average pre-prescription blood hemoglobin A1C level), "visitNo" (number of visits by the patient), and "pastHbA1c1" (past blood hemoglobin A1C level). For example, if a patient has an average pre-prescription blood hemoglobin A1C level of less than 8.582, has at least 4 visits, and has past hemoglobin A1C levels of less than 9.15, then the patient would be prescribed with treatment "4," which corresponds to 0 units of insulin, 1 unit of metformin, and 0 units of oral blood glucose regulation agents. The Optimal Policy Tree (Figure 5) is simpler, but equally interpretable. We observe that it selects different features; for example, the patient's average and past blood hemoglobin A1C levels are not taken into account.

4.4 Real-world deployment potential and challenges

We briefly discuss the potential and challenges in real-world deployment of PNN and Mirrored OCT models. Given the flexibility and ease of training, PNNs and Mirrored OCTs have significant potential for real-world uses. On the other hand, we acknowledge that there are difficulties when it comes to implementing prescriptive models, regardless of the particular model, as well as ethical concerns with using models to make decisions. For example, one challenge in health-care use cases is developing model trustworthiness. However, we believe that the interpretability of Mirrored OCTs, as well as developing models alongside clinician collaborators, makes for promising deployment possibilities. Most importantly, we intent for PNNs and Mirrored OCTs to be used as a second opinion for clinicians; our models not only reinforce clinician decisions when the two opinions match, but our models can also alert clinicians to reviewing their decision when the opinions do not match. In a real-world application, the clinician would make a final decision that leverages their clinical knowledge in conjunction with the recommendation and interpretable features from a Mirrored OCT.

5 Conclusions

With its classification-like feedforward neural network architecture, our PNN framework flexibly handles multimodal data, by easily enabling the incorporation of multiple data sources. Furthermore, it is widely applicable for all treatment scenarios, and has the potential of making a significant impact in a variety of settings, as shown in our extensive real-world experiments. To the best of our knowledge, our method is the first prescriptive model for multimodal data, and it also outperforms or performs comparably to other

well-known prescriptive models on unimodal tabular data in all treatment scenarios, without requiring large computational resources.

Our approach is not only shown to perform strongly quantitatively, but also to provide realistic and stable prescriptions. The discrepancy between the prescribed and real-life treatment distributions is comparable to the other prescriptive methods. The small standard deviation of each sample's assignments from the models indicates that the networks are stable and robust to different data splits. Also, PNNs offer the advantage of flexibility since the user can adjust the loss function to provide partial control to the prescriptions, leveraging expert knowledge.

Deep learning methods generally sacrifice interpretability. On unimodal tabular datasets, we are able to recover interpretability through a knowledge distillation approach leveraging interpretable OCT models, and on multimodal datasets, some interpretability may still be recovered. These Mirrored OCTs demonstrate similarly high performance in our real-world experiments, demonstrating that we can maintain high performance without sacrificing interpretability. This recovery of interpretability is critical for real-world deployment of deep learning models.

We conclude that our multimodal deep learning framework, PNNs, offers both flexibility and strong performance, effectively utilizing deep learning to process multimodal data. By integrating multiple data sources, the framework significantly enhances decision-making capabilities. This unified approach demonstrates its potential as a versatile prescriptive tool, well-suited for a wide range of applications.

References

Naomi Altman. An introduction to kernel and nearest neighbor nonparametric regression. *The American Statistician*, 46(3):175-185, 1992. URL `https://www.jstor.org/stable/2685209`.

Maxime Amram, Jack Dunn, and Ying Daisy Zhuo. Optimal policy trees. *Machine Learning*, 11:2741–2768, 2022.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016. URL `https://doi.org/10.1073/pnas.1510489113`.

Mohsen Bayati, Mark Braverman, Michael Gillam, Karen M. Mack, George Ruiz, Mark S. Smith, and Eric Horvitz. Data-driven decisions for reducing readmissions for heart failure: General methodology and case study. *PLOS ONE*, 9(10), 2014. URL `https://doi.org/10.1371/journal.pone.0109264`.

David Bergman, Teng Huang, Philip Brooks, Andrea Lodi, and Arvind U Raghunathan. Janos: an integrated predictive and prescriptive modeling framework. *INFORMS Journal on Computing*, 34(2):807–816, 2022.

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106:1039–1082, 2017.

Dimitris Bertsimas and Jack Dunn. *Machine Learning under a Modern Optimization Lens*. Dynamic Ideas, 2019.

Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025-1044, 2020. URL `https://pubsonline.informs.org/doi/10.1287/mnsc.2018.3253`.

Dimitris Bertsimas and Nathan Kallus. The power and limits of predictive approaches to observational data-driven optimization: The case of pricing. *INFORMS Journal on Optimization*, 5(1), 2022. URL `https://www.researchgate.net/publication/302568371_Pricing_from_Observational_Data`.

Dimitris Bertsimas, Nathan Kallus, Alexander M Weinstein, and Daisy Ying Zhuo. Personalized diabetes management using electronic medical records. *Diabetes care*, 40(2):210–217, 2017a. URL `https://pubmed.ncbi.nlm.nih.gov/27920019/`.

Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: An optimization approach. *J. Mach. Learn. Res.*, 18:196:1–196:39, 2017b.

Dimitris Bertsimas, Jack Dunn, and Nishanth Mundru. Optimal prescriptive trees. *INFORMS Journal on Optimization*, 1(2):164–183, 2019. URL https://jack.dunn.nz/papers/OptimalPrescriptiveTrees.pdf.

Max Biggs, Wei Sun, and Markus Ettl. Model distillation for revenue optimization: Interpretable personalized pricing, 2021.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. URL https://www.jstor.org/stable/2685209.

Leo Breiman, Jerome Friedman, Charles Stone, and Richard Olshen. Classification and regression trees. *CRC press*, 1984.

William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596-610, 1988. URL https://www.jstor.org/stable/2685209.

Matthew J Colbrook, Vegard Antun, and Anders C Hansen. The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem. *Proceedings of the National Academy of Sciences*, 119(12):e2107151119, 2022.

Miroslav Dudik, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:11034601*, 2011. URL https://arxiv.org/abs/1103.4601.

Egil Ferkingstad, Anders Løland, and Mathilde Wilhelmsen. Causal modeling and inference for electricity markets. *Energy Economics*, 33(3):404-412, 2011. URL https://doi.org/10.1016/j.eneco.2010.10.006.

Homann Georg, Toschke Christina, Gassmann Peter, and Vieth Volker. Accuracy of the aast organ injury scale for ct evaluation of traumatic liver and spleen injuries. *Chinese Journal of Traumatology*, 17(01): 25–30, 2014.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Nathan Kallus. Recursive partitioning for personalization using observational data. *Proceedings of Machine Learning Research*, 70:1789-1798, 2017. URL https://proceedings.mlr.press/v70/kallus17a.html.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Brian K Lee, Justin Lessler, and Elizabeth A Stuart. Weight trimming and propensity score weighting. *PloS one*, 6(3):e18174, 2011.

Pier Pasquale Leone, Andrea Scotti, Edwin C Ho, Manaf Assafin, James Doolittle, Mei Chau, Leandro Slipczuk, Matthew Levitus, Damiano Regazzoli, Antonio Mangieri, et al. Prosthesis tailoring for patients undergoing transcatheter aortic valve implantation. *Journal of Clinical Medicine*, 12(1):338, 2023.

Yikuan Li, Ramsey M. Wehbe, Faraz S. Ahmad, Hanyin Wang, and Yuan Luo. Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences, 2022. URL https://arxiv.org/abs/2201.11838.

Steven Lugauer, Steven Buechler, Timothy Gilbride, and Carrie Heilman. The complete journey, 2020. Data retrieved from dunnhumby, http://www.dunnhumby.com/source-files/.

Andreas Mitsis, Xun Yuan, Christos Eftychiou, Panayiotis Avraamides, and Christoph A. Nienaber. Personalised treatment in aortic stenosis: A patient-tailored transcatheter aortic valve implantation approach. *Journal of Cardiovascular Development and Disease*, 9(11), 2022. ISSN 2308-3425. doi: 10.3390/jcdd9110407. URL https://www.mdpi.com/2308-3425/9/11/407.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Vrishabh Patil, Kara Hoppe, and Yonatan Mintz. Applications of 0-1 neural networks in prescription and prediction. 2024. URL https://arxiv.org/abs/2402.18851.

Scott Powers, Junyang Qian, Kenneth Jung, Alejandro Schuler, Nigam H. Shah, Trevor Hastie, and Robert Tibshirani. Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, 37(11):1767–1787, 2018. doi: 10.1002/sim.7623.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi: https://doi.org/10.1037/h0042519.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. doi: https://doi.org/10.1038/323533a0.

Arman Alam Siddique, Mireille E Schnitzer, Asma Bahamyirou, Guanbo Wang, Timothy H Holtz, Giovanni B Migliori, Giovanni Sotgiu, Neel R Gandhi, Mario H Vargas, Dick Menzies, and Andrea Benedetti. Causal inference with multiple concurrent medications: A comparison of methods and an application in multidrug-resistant tuberculosis. *Stat Methods Med Research*, 28(12):3534–3549, 2019. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6511477/.

Craig R Smith, Martin B Leon, Michael J Mack, D Craig Miller, Jeffrey W Moses, Lars G Svensson, E Murat Tuzcu, John G Webb, Gregory P Fontana, Raj R Makkar, et al. Transcatheter versus surgical aortic-valve replacement in high-risk patients. *New England Journal of Medicine*, 364(23):2187–2198, 2011.

Luis R. Soenksen, Yu Ma, Cynthia Zeng, Leonard Boussioux, Kimberly Villalobos Carballo, Liangyuan Na, Holly M. Wiberg, Michael L. Li, Ignacio Fuentes, and Dimitris Bertsimas. Integrated multimodal artificial intelligence framework for healthcare applications. *NPJ Digital Medicine*, 5, 149(1), 2022. URL https://www.nature.com/articles/s41746-022-00689-4.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Wei Sun and Asterios Tsiourvas. Learning prescriptive ReLU networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33044–33060. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/sun23j.html.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018. doi: 10.1080/01621459.2017.1319839. URL https://doi.org/10.1080/01621459.2017.1319839.

John G. Webb and David A. Wood. Current status of transcatheter aortic valve replacement. *Journal of the American College of Cardiology*, 60(6):483–492, 2012. doi: 10.1016/j.jacc.2012.01.071. URL `https://www.jacc.org/doi/abs/10.1016/j.jacc.2012.01.071`.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. *Operations Research*, 71(1):148–183, 2023. doi: 10.1287/opre.2022.2271.

A Appendix

A.1 PNN Hyperparameters

We specify and finetune the following PNN hyperparameters:

- **Number of layers of the network.** We experiment with both shallow and deep networks. Though conclusions differ based on the dataset, in general we observe that deeper networks do not necessarily improve results.
- **Number of nodes at each layer.** The size of the dataset closely affects this hyperparameter. Typically more nodes per layer are used with larger datasets that also include more features.
- **Batch size.** This parameter determines the number of samples used in each forward pass of the network and for backpropagation, where the network parameters are updated after each batch passes through the network. It also affects the training speed, since too many batches can slow down the training process. Again, there is a correlation between batch size and the size of our data; larger batch sizes are employed for larger datasets.
- **Learning rate.** The learning rate is an important parameter of the training process, since it defines how steep the descent is at each step of the gradient descent algorithm during training. After experimentation for each dataset, we find an appropriate learning rate that is not too big so that the algorithm becomes stuck in local optima but also not too small so that convergence is too slow.
- **Weight decay.** This hyperparameter scales an L_2-regularization term of the network weights that is added to the objective function to prevent them from taking too large values. Since our data is normalized, we observe that lowering the weight decay coefficient and relaxing the weights are actually beneficial and do not result in overfitting.
- **Number of epochs.** The number of epochs is a particularly hard parameter to tune, since we want to prevent overfitting but also allow the training to continue until sufficient convergence. For this reason, we employ early stopping, a technique that is adaptive to each specific training process and terminates training when a fluctuation in the validation loss is observed. Such a fluctuation indicates that the network is no longer improving in out-of-sample data generalization.

A.2 Additional tables

A.3 Extension of interpretability of unstructured and structured real-world datasets

Liver trauma. We discuss the multimodal Mirrored OCT from Figure 3 for the liver injury dataset. The embedding features from the clinical notes are not nearly as interpretable, but we can still interpret the

Table 10: Treatment scenarios, data splits, and counterfactual estimation methods used for each structured dataset.

Dataset	**Treatment scenario**	**Counterfactual estimator**	**Training Accuracy(%) of Mirrored OCT**
Groceries pricing	Single continuous	XGBoost Classifier (direct method)	98.03 ± 0.81
Splenic injuries treatment	Multiple discrete	Random Forest Classifier (doubly-robust)	91.99 ± 1.65
REBOA in blunt trauma patients	Binary	XGBoost Classifier (doubly-robust)	96.93 ± 0.69
Diabetes management	Multiple continuous	Random Forest Regressor (doubly-robust)	91.54 ± 3.07

tabular features selected by the Mirrored OCT. We see that the OCT selects two note features, as well as whether the patient had other symptoms with the circulatory or respiratory system and whether the patient had pain in their throat or chest.

Groceries. We compare the resulting Mirrored OCT from Figure 6 and the Optimal Policy Tree (OPT) from Figure 7. The features selected by both the Mirrored OCT and the Optimal Policy Tree are similar, with the most prominent ones being the homeowner status, age, income range, household status, and marital status. An interesting difference between the two trees is the number of distinct prescriptions selected: the OCT only selects 3 of the 6 pricing options – prediction classes 0, 1, and 5 (which correspond to prices USD \$2.00, \$2.50, and \$5.00) – whereas the OPT prescribes more of the possible options. The OCT's strategy seems to therefore select low prices for lower-income households, compared to high prices for households with more financial stability.

Splenic injuries treatment. We compare the Mirrored OCT displayed in Figure 8 with an Optimal Policy Tree (OPT) in Figure 9. There are three possible treatments: simple observation (treatment 0 of the OCT, "a" for the OPT), splenectomy (treatment 1 of the OCT, "b" for the OPT) and angioembolization (treatment 2 of the OCT, "c" for the OPT). The OCT prescribes the first and third options while the OPT prescribes all three. The two trees split on similar features, although at different levels of the tree; these features include "totalgcs" (Glasgow coma scale), the state (grade) of the spleen (injury severity on a scale of increasing acuteness from 1 to 5), diabetes, and smoking status. However, the Mirrored OCT splits on SBP (systolic blood pressure), TBI (Traumatic brain injury) and intubation status, while the OPT splits on BMI, respiratory rate, and history of cirrhosis.

REBOA in blunt trauma patents. We compare the Mirrored OCT from Figure 10 and the Optimal Policy Tree (OPT) from Figure 11.

From Figure 10, we clearly see that the important features include, among others, "pneumothorax_1" (whether the patient has pneumothorax, a collection of air outside the lung but within the pleural cavity), "gcs" (Glasgow coma scale), SBP (systolic blood pressure), and pulse rate. We observe that the OPT (Figure 11) only splits on SBP and pulse rate, indicating that the resulting Mirrored OCT is considering more factors to make the final prescription, which is potentially closer to real life scenarios.

A.4 Mirrored OCTs on all real-world datasets

Further examples of Mirrored OCTs of maximum depth 7 for the TAVR, liver trauma, diabetes, groceries, splenic injuries, and REBOA datasets can be found under `https://drive.google.com/drive/folders/12XNOQllyzVQEFFguHvkcAQ-7Psp9KriQ`.

Figure 2: Mirrored OCT of maximum depth 7 for TAVR.

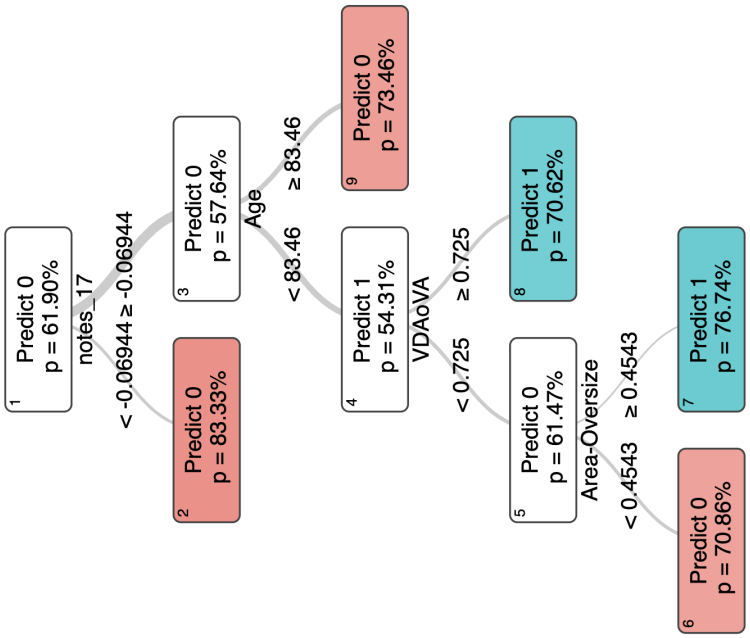


Figure 3: Mirrored OCT of maximum depth 7 for liver trauma.

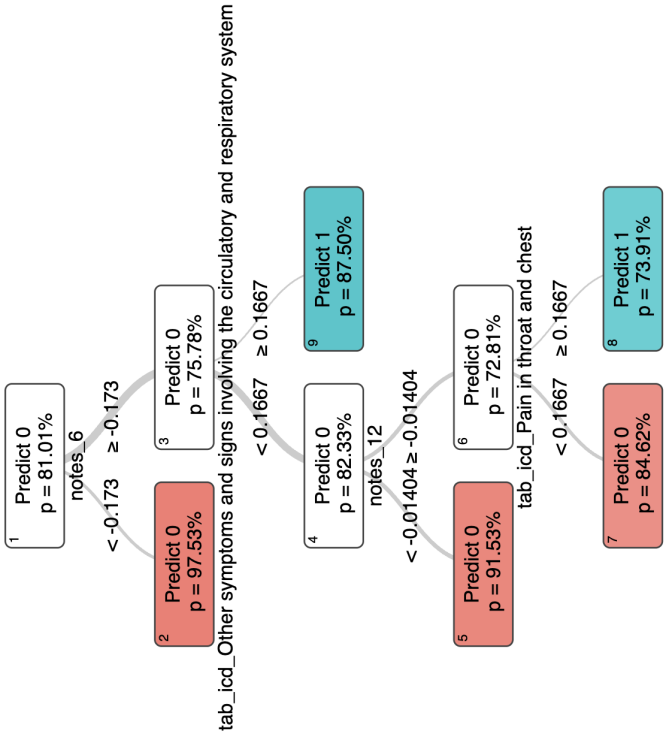
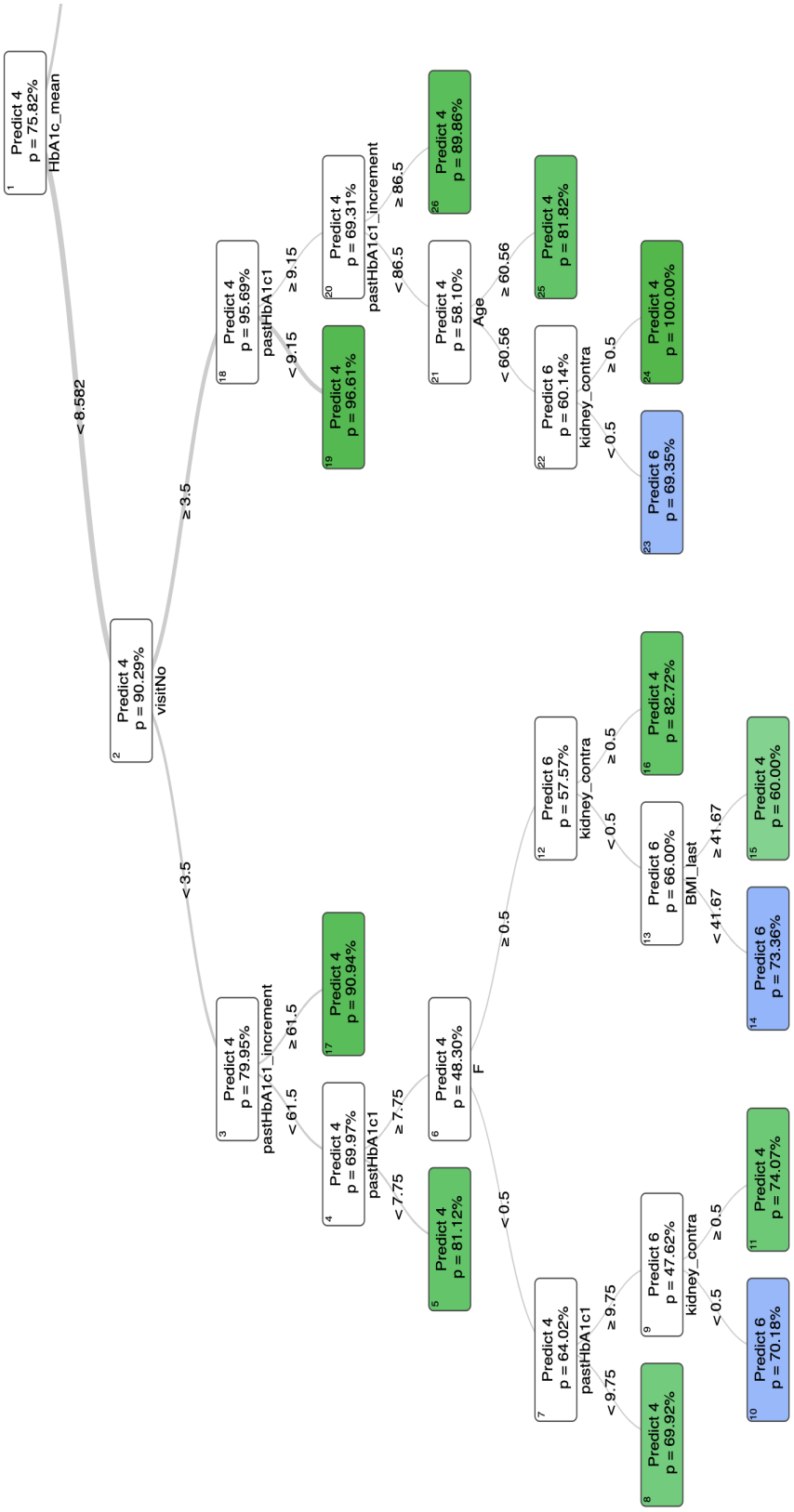


Figure 4: Mirrored OCT of maximum depth 7 for diabetes management.



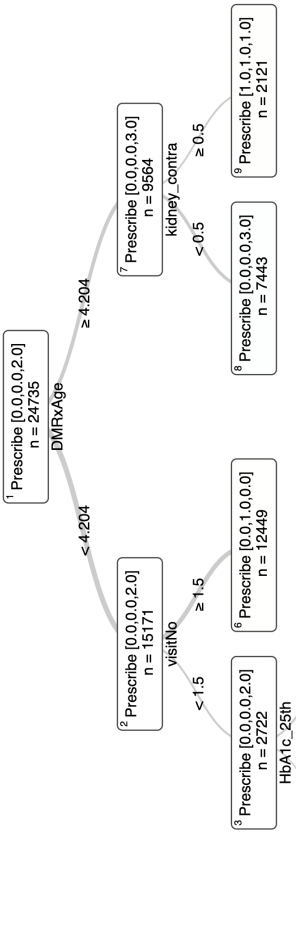


Figure 5: Optimal Policy Tree of maximum depth 7 for diabetes management.

Figure 6: Mirrored OCT of maximum depth 7 for groceries.

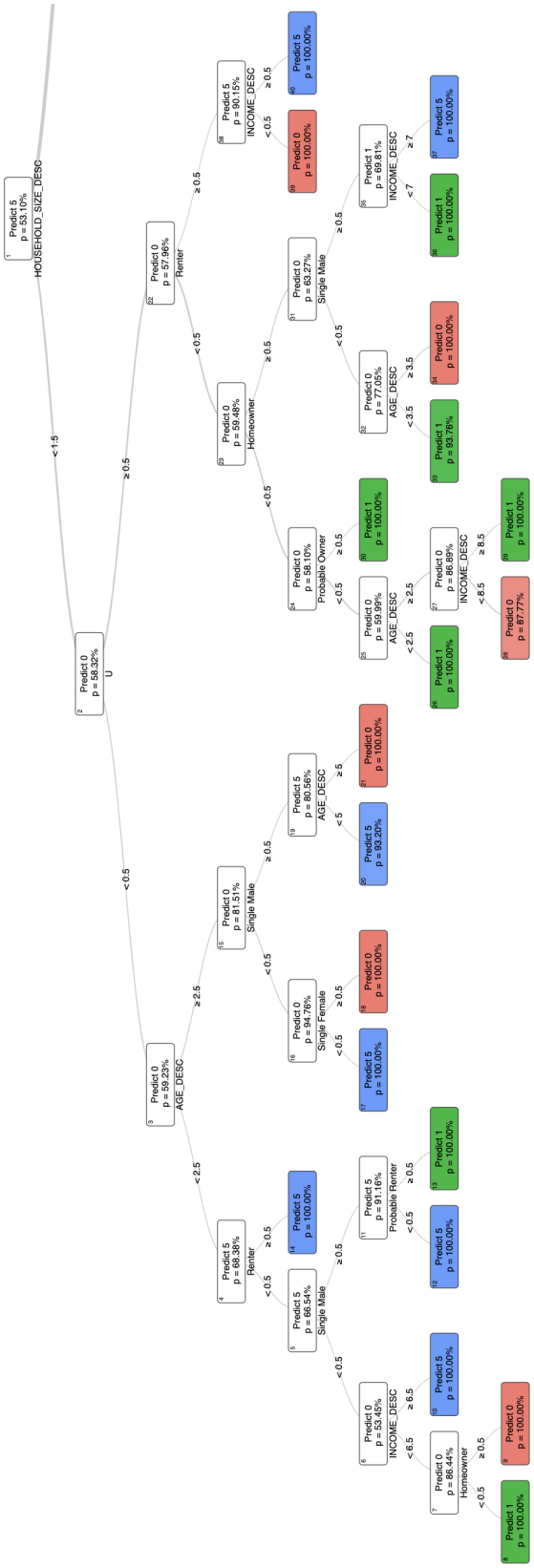


Figure 7: Optimal Policy Tree of maximum depth 7 for groceries.

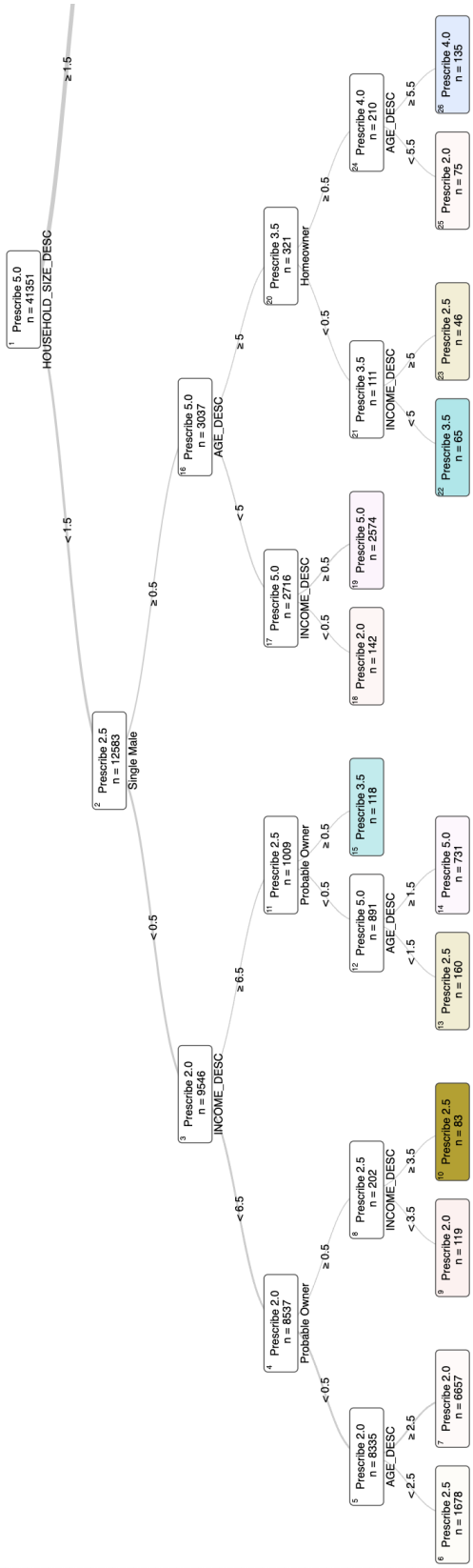


Figure 8: Mirrored OCT of maximum depth 7 for spleen injury treatment.

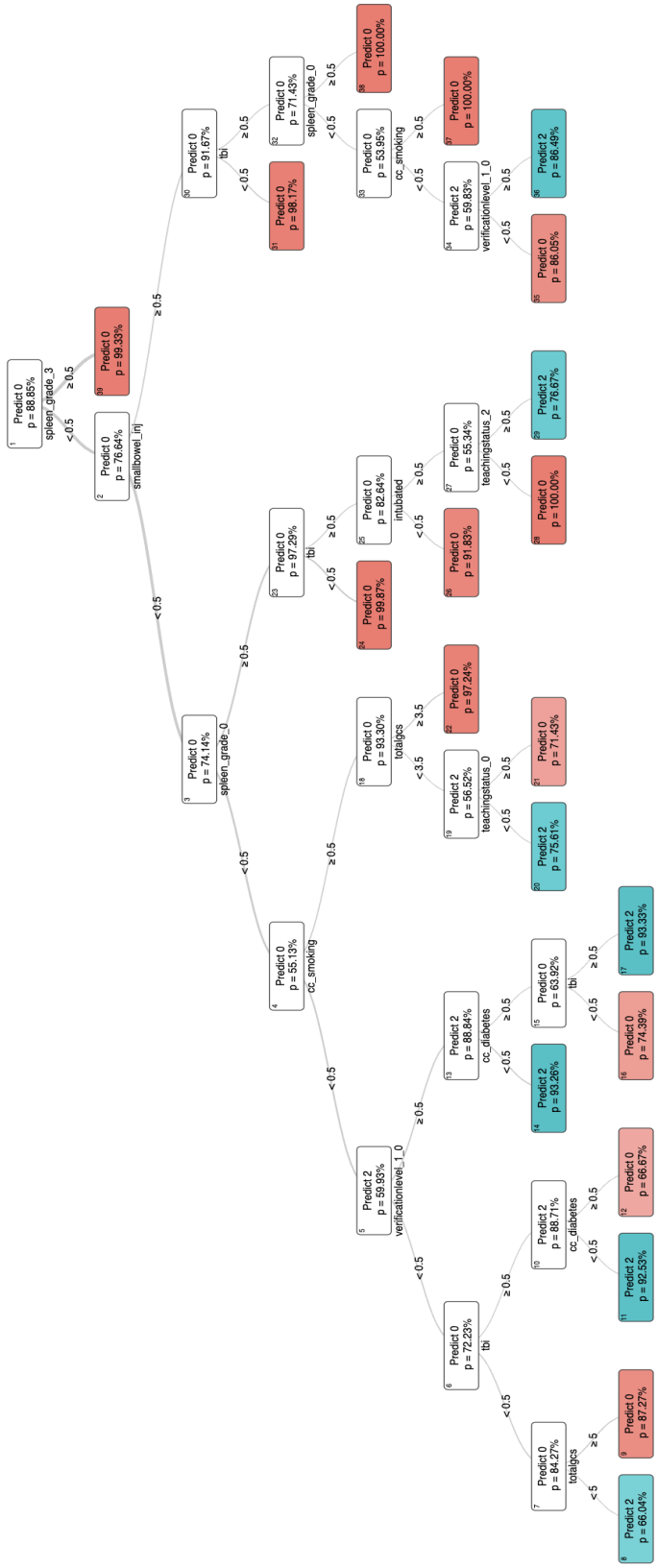


Figure 9: Optimal Policy Tree of maximum depth 7 for spleen injury treatment.

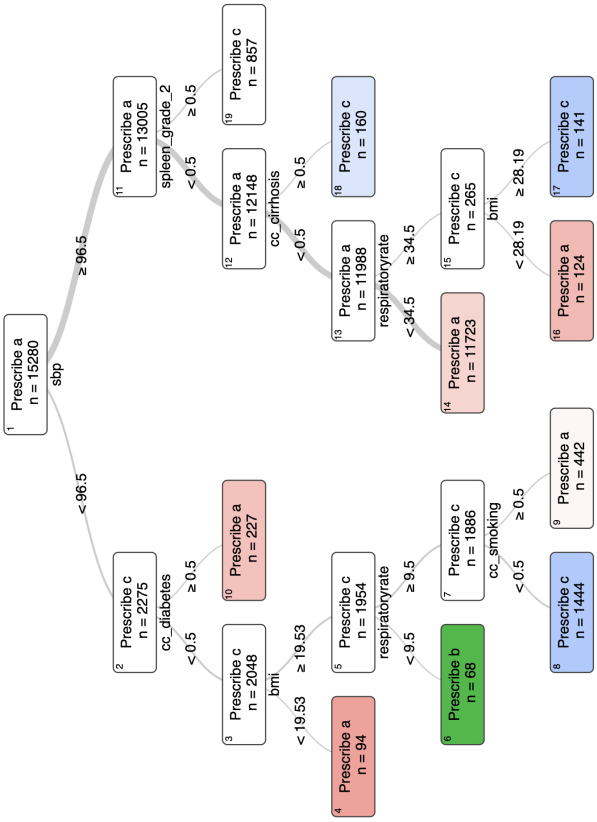
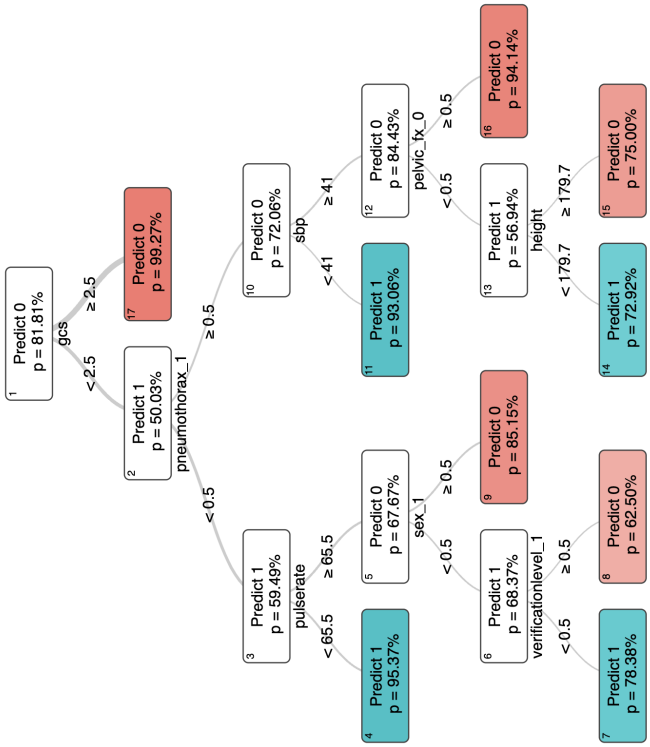


Figure 10: Mirrored OCT of maximum depth 7 for REBOA in blunt trauma patients.



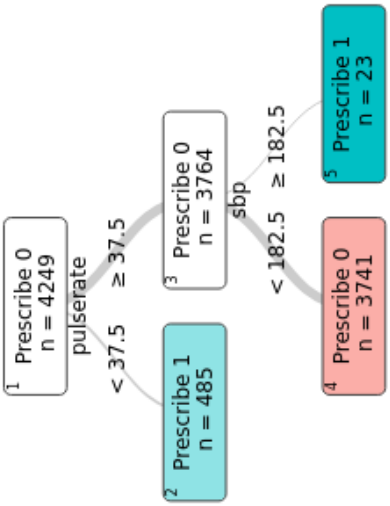


Figure 11: Optimal Policy Tree of maximum depth 7 for REBOA in blunt trauma patients.