Towards Certainty: Exploiting Monotonicity with Fast Marching Methods to Reduce Predictive Uncertainty

Anonymous authors
Paper under double-blind review

Abstract

In recent years, neural networks have achieved impressive performance on a wide range of tasks. However, neural networks tend to make overly optimistic predictions about out-of-distribution data. When managing model risks, it is important to know what we do not know. Although there have been many successes in detecting out-of-distribution data, it is unclear how we can extract further information from these uncertain predictions. To address this problem, we propose to use three types of monotonicity to extract further information by solving a mean-variance optimization problem. The fast marching method is proposed as an efficient solution. We demonstrate, using empirical examples, that it is possible to provide confident bounds for a large portion of uncertain predictions by monotonicity.

1 Introduction

Deep neural networks (DNNs) have been widely applied in a variety of applications. For high-risk sectors such as the financial sector, predictive power is not the only factor to consider. It is stressed in the model risk management handbook [1] provided by the Office of the Comptroller of the Currency (OCC) [2] that when machine learning (ML) models are applied, they need to **know what they don't know**. It is possible, for example, for a trading model to perform very well under normal circumstances, but to fail during a financial crisis. To prevent losing money, traders might switch to other strategies if the trading model alerts them to significant changes in the trading environment.

However, in practice, the distribution of the population may differ significantly from that of the training set. For instance, in lending, there is an abundance of data on good applicants, but a very risky applicant with numerous past dues might be extremely rare and have not been seen by the model before. Out-of-distribution (OOD) inputs often pose a challenge to the use of DNNs. Unless carefully managed, DNNs may become overconfident about their predictions, resulting in catastrophic results.

There has been considerable attention given to this challenge in recent years, and the results are encouraging (Lakshminarayanan et al., 2017; Kardan et al., 2021; Bibas et al., 2021; Liang et al., 2018; Yang et al., 2022; Geng et al., 2020; Zhou et al., 2022; Ovadia et al., 2019). Based on the successful detection of OOD data, we ask the following question: **What can we learn from what we don't know?**

As with black-box ML models, we may be unable to obtain additional information. On the other hand, recent developments in **domain-knowledge-inspired machine learning models** have been highly successful and could be used to provide additional information. In particular, **monotonic machine learning models** has been very popular in many areas (Repetto, 2022; Chen & Ye, 2022; Liu et al., 2020; Chen & Ye, 2023). There may be additional information to be gained from monotonic models in the case of uncertain predictions. It has been shown by Chen (2022) that individual monotonicity can provide reliable bounds to uncertain inputs.

[1] https://www.occ.treas.gov/publications-and-resources/publications/comptrollers-handbook/files/model-risk-management/index-model-risk-management.html.

[2] The Office of the Comptroller of the Currency charters, regulates, and supervises all national banks, federal savings associations, and federal branches and agencies of foreign banks. In general, the Office of the Comptroller of the Currency provides regulatory guidance to financial institutions. In their handbook, they provide practitioners with a high-level guide to controlling model risk and validating AI and ML models in financial markets.

Here is an example of a simple illustration in credit scoring. Credit scores are based on information in credit reports to predict credit behavior, including the likelihood of repaying a loan in a timely manner. ML models can be used to predict the probability of default, which are then converted into credit scores. Predictions are based on a number of factors, one of which is the number of past-due payments. Let's suppose that an applicant has ten past-due payments. Although the model may not be certain of this prediction due to its rarity, it does know that five past-due payments has already been very risky, so ten past-due payments cannot be any less risky. Consequently, it should be categorized into risky groups.

There has been significant attention paid to monotonicity in the past (Sill, 1997; Cano et al., 2019; Gupta et al., 2020), since it is not only about conceptual soundness but also about **fairness**. In the case of credit scoring when individual monotonicity is involved, if an applicant has one more past-due payment, the model should then predict that the probability of default is increasing based on the conceptual soundness and fairness outlined in the OCC's handbook. Otherwise, the model would be unfair since an additional past-due payment has not been penalized. Thus, monotonicity is usually a **hard requirement** for related applications.

While individual monotonicity has been extensively studied (Liu et al., 2020; Milani Fard et al., 2016; You et al., 2017; Runje & Shankaranarayana, 2023), **pairwise monotonicity** has been largely ignored. Recent studies (Gupta et al., 2020; Cotter et al., 2019; Chen & Ye, 2022; 2023) have shown that pairwise monotonicity is also important. The idea behind pairwise monotonicity is that some features are intrinsically more important than others. For example, in credit scoring, past-due payment information can be divided into two features based on the number of past-due payments within three months or more than three months. It is then important to consider the feature that counts the number of past-due payments of more than three months as more important for fairness. Alternatively, if the ML model predicts an applicant with a past-due payment within three months is more risky than another with a past-due payment of more than three months, then the prediction is unfair.

It is possible to provide more information for models containing pairwise monotonicity. For example, if an ML model is sure that an applicant with three past-due payments less than three months is already very risky, then an applicant with three past-due payments greater than three months should only be more risky, even if the model is unsure of the specific predictions it makes. Using this idea, we generalize the mean-variance optimization problem proposed by Chen (2022). This results in a complex **non-convex mixed-integer nonlinear programming (MINLP)** problem. Generally, such a problem is difficult to solve, as discussed in Burer & Letchford (2012). Alternatively, by taking advantage of the monotonic property of models, we propose to use the **Fast Marching Method for Optimization (FMMO)** to find the **global** optimizer to the problem. The FMMO algorithm is inspired by the classical Fast Marching Method for Eikonal equation (FMME) for tracing interface evolution in numerical partial differential equations (Tsitsiklis, 1995; Sethian, 1996; Helmsen et al., 1996). By utilizing general types of monotonicity, we extend the fast marching method to solve our optimization problem. By using empirical examples, we demonstrate that our method has the capability of providing reliable bounds to unconfident predictions and enhances the prediction of uncertainty.

In this work, we make three major contributions:

- We generalize the two-stage framework by Chen (2022) with only individual monotonicity to general types of monotonicity. By incorporating pairwise monotonicity, it will be possible to search a larger search space for optimization, resulting in better global optimizers.

- The monotonicity-induced optimization geometry of the domain is studied, providing an intuitive understanding of the geometry and permitting the implementation of algorithms in practice.

- The FMMO based on monotonicity is proposed to find the **global** optimizer to the complex **non-convex mixed-integer nonlinear programming** optimization. Empirical results indicate that the FMMO with all monotonicity has improved the accuracy of the baseline method.

2 Prerequisites

For problem setup, assume we have n samples $\{\mathbf{x}_i\}_{i=1}^n$ and m features such that $\mathbf{x}_i \in \mathbb{R}^m$, the data-generating process is

$$y_i = f(\mathbf{x}_i) + \epsilon_i \tag{1}$$

for regression problems , where $\mathbf{x}_i$ is a random input, y_i is a label, f is the ML model, and ϵ_i is the random noise, and

$$y_i|\mathbf{x}_i = \text{Bernoulli}(g^{-1}(f(\mathbf{x}_i))) \tag{2}$$

for binary classification problems, where g is the link function (e.g., logistic function). For simplicity, we assume $x_j \in \mathbb{R}^+ \cup \{0\}$ where x_j denotes the jth feature of $\mathbf{x}$. Assumptions of this type are common in cost-sharing problems (Friedman & Moulin, 1999) and are often reasonable for high-stakes applications. Then ML methods are applied to approximate f.

2.1 Individual and Pairwise Monotonicity

Monotonicity is crucial for ensuring conceptual soundness and fairness and is therefore often strictly required in high-stakes applications (Chen & Ye, 2023; Gupta et al., 2020; Liu et al., 2020). Throughout the paper, without loss of generality (WLOG), we focus on the monotonically increasing functions. Suppose f is individual monotonic with respect to x_α and we partition $\mathbf{x} = (x_\alpha, \mathbf{x}_\neg)$. The well-known **individual monotonicity** is then defined as below.

Definition 2.1. *We say f is* ***individually monotonic*** *with respect to (w.r.t) x_α if $\forall x_\alpha, \mathbf{x}_\neg$, $\forall c \in \mathbb{R}^+$*

$$f(x_\alpha, \mathbf{x}_\neg) \leq f(x_\alpha + c, \mathbf{x}_\neg). \tag{3}$$

Individual monotonicity is a common phenomenon in practice. In credit scoring, for instance, the probability of default should be individually monotonic with respect to the number of past dues. In criminology, the likelihood of recidivism is individually monotonic with respect to the number of past criminal charges. In education, the probability of admission should be monotonic in relation to a student's grade point average.

In practice, certain features are intrinsically more important than others, and **pairwise monotonicity** (Gupta et al., 2020; Cotter et al., 2019; Chen & Ye, 2023; 2022) describes these phenomena. Analog to equation 3, we partition $\mathbf{x} = (x_\beta, x_\gamma, \mathbf{x}_\neg)$. WLOG, we assume x_β is more important than x_γ. In addition, we require all features with pairwise monotonicity also satisfy individual monotonicity. There are two types of pairwise monotonicity. We start with the **strong pairwise monotonicity**.

Definition 2.2. *We say f is* ***strongly monotonic*** *w.r.t x_β over x_γ if $\forall x_\beta, x_\gamma, \mathbf{x}_\neg, \forall c \in \mathbb{R}^+$*

$$f(x_\beta, x_\gamma + c, \mathbf{x}_\neg) \leq f(x_\beta + c, x_\gamma, \mathbf{x}_\neg). \tag{4}$$

As an example, in criminology, for each additional crime, a felony is considered more serious than a misdemeanor for predicting the probability of recidivism. Next, we introduce the **weak pairwise monotonicity**.

Definition 2.3. *We say f is* ***weakly monotonic*** *w.r.t x_β over x_γ if $\forall x_\beta, x_\gamma$ s.t. $x_\beta = x_\gamma, \forall \mathbf{x}_\neg, \forall c \in \mathbb{R}^+$,*

$$f(x_\beta, x_\gamma + c, \mathbf{x}_\neg) \leq f(x_\beta + c, x_\gamma, \mathbf{x}_\neg). \tag{5}$$

When it comes to predicting admission acceptance for STEM majors, math scores on the GRE are more important than verbal scores. In contrast to strong pairwise monotonicity, such comparisons are not always valid. If a student already has a good math score but a very poor verbal score, then an additional increase in verbal scores might increase more chances than math since universities want to ensure the student is capable of communicating effectively. The condition $x_\beta = x_\gamma$ ensures that the comparison is made at the same magnitude since the comparison is not always valid. The result is that, out of 170 points for math and verbal, a student with math 165 and verbal 150 has a greater chance of admission for STEM majors than a student with math 150 and verbal 165, but not necessarily a greater chance than a student with math 160 and verbal 155.

2.2 Detect Out-of-Distribution Data

Ovadia et al. (2019) presents a large-scale evaluation of different methods for quantifying predictive uncertainty across a variety of data modalities and architectures. Overall, they found that the ensemble methods by Lakshminarayanan et al. (2017) performed the best across most metrics and were the most robust to data shifts. Following Lakshminarayanan et al. (2017), models are trained M times with random initialization and data shuffles in the entire dataset with $\{\widehat{f}(\mathbf{x};\boldsymbol{\theta}_i)\}$, where $\{\boldsymbol{\theta}_i\}_{i=1}^M$ denote the parameters of each neural network in the ensemble. For the prediction, the average of the ensembles is used,

$$\widehat{\mu}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}\widehat{f}(\mathbf{x};\boldsymbol{\theta}_i). \tag{6}$$

The variance is used as a proxy for the level of model uncertainty and is calculated by

$$\widehat{\sigma}^2(\mathbf{x}) = \frac{1}{M-1}\sum_{i=1}^{M}(\widehat{f}(\mathbf{x};\boldsymbol{\theta}_i) - \widehat{\mu}(\mathbf{x}))^2. \tag{7}$$

For a data point $\mathbf{x}$, when the variance is large, say $\widehat{\sigma}^2(\mathbf{x}) > \epsilon$ for a threshold ϵ, it is considered too risky to make the prediction. Accordingly, the dataset is divided into confident and unconfident sets.

3 Two-stage Method

We generalize the two-stage framework presented by Chen (2022) for general monotonicity. As a first step, using ensemble methods, we identify the unconfident set for OOD data. In the second stage, we solve a mean-variance optimization problem to provide bounds for the points in the unconfident set. Using general monotonicity, the search domain can be enlarged. This leads to a better optimizer, however, the problem becomes more complex with a larger domain, and more importantly, the geometry becomes more complex, possibly non-convex. Lastly, if bounds do not meet expectations, we leave them to human judgment.

3.1 Stage I: Detect Out-of-Distribution Data

To detect the unconfident set, we wish to utilize the ensemble method. We demonstrate in the following Theorem that monotonicity is preserved by ensembles. In this regard, we can choose to use the ensemble $\widehat{\mu}(\mathbf{x})$ as the prediction.

Proposition 3.1. *If $\widehat{f}(\mathbf{x};\boldsymbol{\theta}_i)$ achieves all monotonicity for all i, then $\widehat{\mu}$ preserves all monotonicity.*

The proof is left in Appendix B. We then apply the ensemble method and consider

$$\mathbb{S} = \{\mathbf{x}|\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon\} \tag{8}$$

as the **unconfident set**. Similarly, we define $\mathbb{Q} = \{\mathbf{x}|\widehat{\sigma}^2(\mathbf{x}) < \epsilon\}$ as the confident set. A vertical bar | used in the set-builder notation throughout the manuscript denotes its meaning as "such that". In this example, $\mathbb{S}$ is the set of all samples $\mathbf{x}$ such that $\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon$.

3.2 Stage II: Mean-Variance Optimization

Predictions with low confidence are generally be excluded from decision making (Ovadia et al., 2019; OCC, 2021). We wish to provide more information about the unconfident predictions once the unconfident set has been determined. For an unconfident prediction $\mathbf{x} \in \mathbb{S}$ such that $\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon$, it would be helpful to provide confident bounds in order to provide more information. Specifically, we wish to find $\mathbf{x}'$ and $\mathbf{x}''$ such that

$$\widehat{\mu}(\mathbf{x}) \in [\widehat{\mu}(\mathbf{x}'), \widehat{\mu}(\mathbf{x}'')], \text{ with } \widehat{\sigma}^2(\mathbf{x}') < \epsilon, \widehat{\sigma}^2(\mathbf{x}'') < \epsilon. \tag{9}$$

However, since we are unconfident about the prediction $\widehat{\mu}(\mathbf{x})$, we cannot directly compare $\widehat{\mu}(\mathbf{x})$ with $\widehat{\mu}(\mathbf{x}')$ and $\widehat{\mu}(\mathbf{x}'')$ for general models. For models that have been assumed to have monotonicity from domain knowledge, it is possible to draw comparisons directly from the monotonicity assumption. For example, in credit scoring, the probability of default should be monotonically increasing w.r.t the number of past-due payments to penalize late payments. Thus, we know that an applicant with four past-due payments will have a greater default probability than an applicant with three past-due payments, assuming all other conditions are equal. Similarly, in criminology, the probability of recidivism should be monotonically increasing w.r.t the number of past criminal changes.

Definition 3.2. *We define the space* $\Omega(\mathbf{x})$ *as*

$$\Omega(\mathbf{x}) = \{\mathbf{x}' | f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})\}. \tag{10}$$

whereas $\underset{M}{\leq}$ *denotes the* ***inequality by monotonicity****. That is, we know* $f(\mathbf{x}') \leq f(\mathbf{x})$ *by the assumed monotonicity from the Definition 2.1, 2.2, 2.3. Similarly,* $\underset{M}{\geq}$ *can be defined.*

Example 3.3. *Here is a demonstration example from the credit scoring. Assume* f *is the probability of default and* x_1, x_2 *is the number of past-due payments greater than three months and less than three months. Then* f *is individually monotonic w.r.t* x_1, x_2 *and strongly monotonic w.r.t* x_1 *over* x_2*. For simplicity, let's focus only on the domain* $0 \leq x_1 + x_2 \leq 2$*. Suppose now we have trained several neural networks with mean* $\widehat{\mu}$ *and variance* $\widehat{\sigma}^2$*. Assume that we are interested in* $\mathbf{x} = (1, 0)$ *with* $\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon$*, therefore* $\widehat{\mu}(\mathbf{x})$ *could be unreliable. We know that* $f(0, 0) \leq f(1, 0)$ *based on the individual monotonicity; we further know* $f(0, 1) \leq f(1, 0)$ *based on strong pairwise monotonicity. This gives us* $\Omega(\mathbf{x}) = \{(0, 0), (0, 1)\}$ *such that* $f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})$ *for* $\mathbf{x}' \in \Omega(\mathbf{x})$*. Although we don't know exactly what* f *exactly is, we can determine* $\Omega(\mathbf{x})$ *solely based on monotonicity. We emphasize the use of* $\underset{M}{\leq}$ *on* f *rather than the outputs from the function* $\widehat{\mu}$*. For example,* $\widehat{\mu}$ *may yield that* $\widehat{\mu}(1, 0) \leq \widehat{\mu}(0, 2)$*. However, we may find that this inequality is not reliable if we are uncertain about the* $\widehat{\mu}(1, 0)$*.*

We wish to provide confident bounds $[\widehat{\mu}(\mathbf{x}'), \widehat{\mu}(\mathbf{x}'')]$ as tight as possible to provide more information. **WLOG, we focus on finding the lower bound $\widehat{\mu}(\mathbf{x}')$, but finding the upper bound $\widehat{\mu}(\mathbf{x}'')$ is similar.** Specifically, we wish to maximize $\widehat{\mu}(\mathbf{x}')$ with $\widehat{\sigma}^2(\mathbf{x}') < \epsilon$, similar to the modern portfolio theory by Markowitz (1952). In summary, for each $\mathbf{x} \in \mathbb{S}$, we wish to solve the following optimization problem

$$\begin{cases} \max_{\mathbf{x}' \in \Omega(\mathbf{x})} \widehat{\mu}(\mathbf{x}'), \\ \text{subject to } \widehat{\sigma}^2(\mathbf{x}') < \epsilon, \\ \Omega(\mathbf{x}) = \{\mathbf{x}' | f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})\}. \end{cases} \tag{11}$$

Although we can obtain the maximum value of $\widehat{\mu}$, it may not be useful if $\widehat{\mu}$ is too small. In practice, we would focus only on $\widehat{\mu} \geq \tau$, for some τ determined by users based on their risk appetites. We may be unable to find satisfactory lower bounds, in which case we leave the decision to human judgment. This may be the case, for example, if we have outliers for nonmonotonic features for which we lack domain expertise. Furthermore, if the variance of all points in the domain is high, there may be no solution. These data points are considered to be part of the **undecided set** $\mathbb{V}$ such that

$$\begin{aligned} \mathbb{V}(\tau) = \{\mathbf{x} | \mathbf{x} \in \mathbb{S} \text{ and } \min_{\mathbf{x}' \in \Omega(\mathbf{x})} \widehat{\sigma}^2(\mathbf{x}') > \epsilon\} \\ \cup \{\mathbf{x} | \mathbf{x} \in \mathbb{S} \text{ and } \mathbf{x} \text{ solves equation 11 and } \widehat{\mu}(\mathbf{x}) < \tau\}. \end{aligned} \tag{12}$$

Similarly, we define the **decided set** $\mathbb{W}$ as

$$\mathbb{W}(\tau) = \{\mathbf{x} | \mathbf{x} \in \mathbb{S} \text{ and } \mathbf{x} \text{ solves equation 11 and } \widehat{\mu}(\mathbf{x}) \geq \tau\}. \tag{13}$$

A demonstration of the two-stage framework is provided in Figure 1.

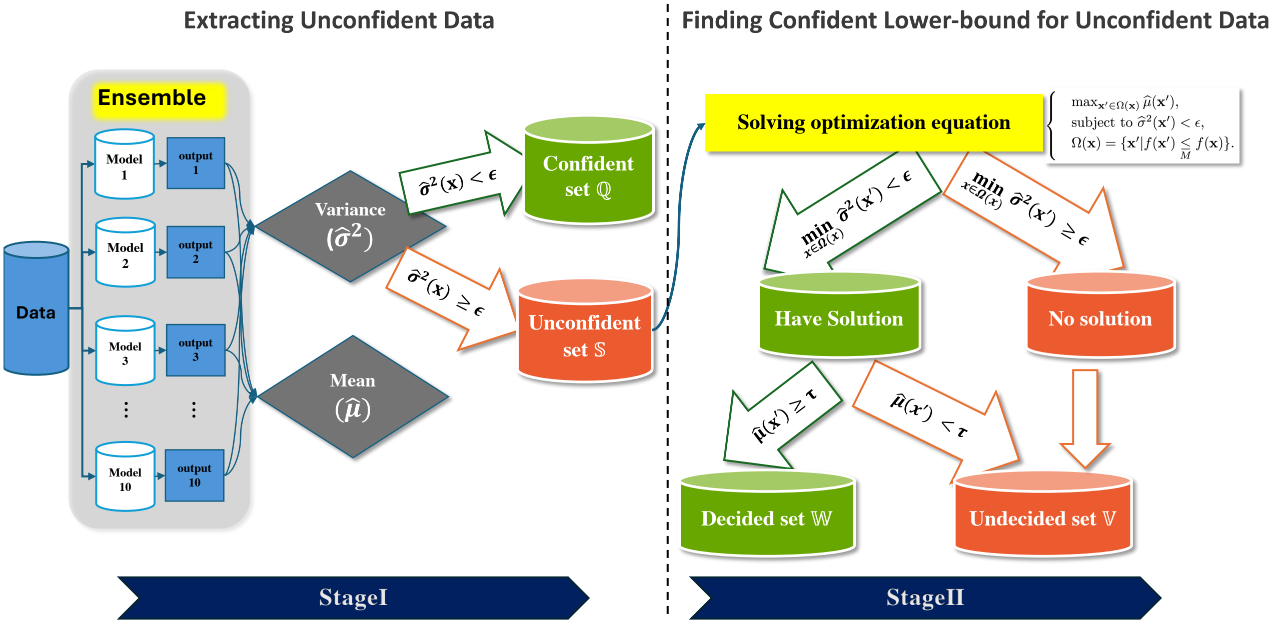

Extracting Unconfident Data
Finding Confident Lower-bound for Unconfident Data
Ensemble
Data
Model 1
Model 2
Model 3
Model 10
output 1
output 2
output 3
output 10
Variance ($\widehat{\sigma}^2$)
Mean ($\widehat{\mu}$)
$\widehat{\sigma}^2(\mathbf{x}) < \epsilon$
$\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon$
Confident set $\mathbb{Q}$
Unconfident set $\mathbb{S}$
Solving optimization equation
$\max_{\mathbf{x}' \in \Omega(\mathbf{x})} \widehat{\mu}(\mathbf{x}')$, subject to $\widehat{\sigma}^2(\mathbf{x}') < \epsilon$, $\Omega(\mathbf{x}) = \{\mathbf{x}' | f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})\}$.
$\min_{x \in \Omega(x)} \widehat{\sigma}^2(x') < \epsilon$
$\min_{x \in \Omega(x)} \widehat{\sigma}^2(x') \geq \epsilon$
Have Solution
No solution
$\widehat{\mu}(x') \geq \tau$
$\widehat{\mu}(x') < \tau$
Decided set $\mathbb{W}$
Undecided set $\mathbb{V}$
StageI
StageII

Figure 1: Two-stage Method

4 Monotonicity-induced Geometry

We would like to provide a more explicit form for $\Omega(\mathbf{x})$ for a better understanding of the geometry, which also permits us to implement the algorithm in practice. We ignore nonmonotonic features for the sake of simplicity unless otherwise stated since we cannot draw any conclusions from them. The features are divided into individual monotonic, weak pairwise monotonic, and strong pairwise monotonic parts, as $\mathbf{x} = (\mathbf{x}_S, \mathbf{x}_U, \mathbf{x}_P)$. For features with weak pairwise monotonicity in U, we give two lists $\mathbf{u}$ and $\mathbf{v}$ with $|\mathbf{u}| = |\mathbf{v}|$ such that f is weakly pairwise monotonic with respect to x_{u_i} over x_{v_i} for $i = 1, \ldots, |\mathbf{u}|$. For strong pairwise monotonicity, we assume that there is a list $\mathbf{p}$ such that f is strongly pairwise monotonic to x_{p_j} over $x_{p_{j+1}}$ for $j = 1, \ldots, |\mathbf{p}| - 1$. All monotonic features follow individual monotonicity. This structure is sufficient for most applications, but more complicated structures can be considered if necessary.

For ease of reading, we provide the final result and leave technical details in Appendix A and B. We find it more convenient to identify the domain boundary first. We consider the maximum boundary points to provide us with points at the boundary of the domain with the largest magnitudes.

Definition 4.1. *The set of maximum boundary points of Ω is defined as follows:*

$$\partial\Omega(\mathbf{x}) = \left\{ \mathbf{x}' \middle| \sum_{i=1}^{m} x'_i = \sum_{i=1}^{m} x_i, f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x}) \right\}. \tag{14}$$

We are now ready to provide the expression for the geometry. Once the maximum boundary points are known, the geometry becomes clear.

Theorem 4.2. *With assumptions in Section 4, The geometry of the domain and the corresponding maximum boundary points are*

$$\partial\Omega(\mathbf{x}) = \mathbf{x} \cup \varphi(\mathbf{x}, \mathbf{p}) \cup \bigcup_{i: x_{u_i} > x_{v_i}} \partial\Omega(\mathbf{x}' | \mathbf{x}' = \Gamma(\mathbf{x}, u_i, v_i)), \tag{15}$$

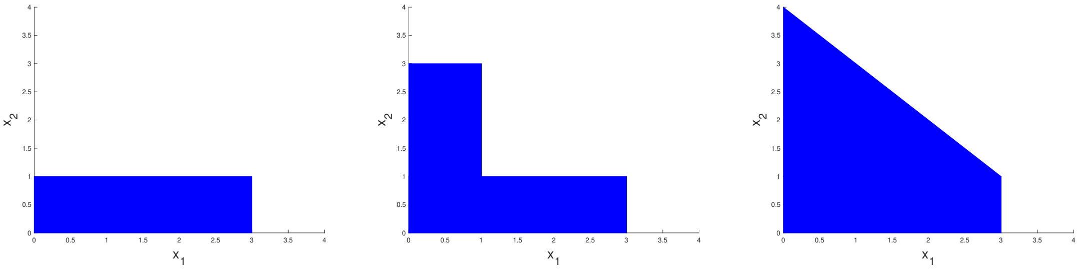
$$\Omega(\mathbf{x}) = \{\mathbf{x}' | \exists \mathbf{x}'' \in \partial\Omega(\mathbf{x}) \ s.t. \ \mathbf{x}' \leq \mathbf{x}''\}, \tag{16}$$

where

$$\Gamma(\mathbf{x},\beta,\gamma)=\begin{cases}x_i & \textit{if } i\neq\beta \textit{ and } i\neq\gamma,\\ x_\gamma, & \textit{if } i=\beta,\\ x_\beta, & \textit{if } i=\gamma.\end{cases} \tag{17}$$

$$\varphi(\mathbf{x},\mathbf{p})=\left\{\mathbf{x}'\middle| x'_i\leq\sum_{j=1}^{i}x_j-\sum_{j=1}^{i-1}x'_j,\forall i=1,\dots,|\mathbf{p}|,\sum_{i=1}^{|\mathbf{p}|}x_i=\sum_{i=1}^{|\mathbf{p}|}x'_i \textit{ and } x'_j=x_j,\forall j\notin\mathbf{p}\right\}. \tag{18}$$

We provide an example of geometry for $\mathbf{x}=(3,1)$ for individual, weak pairwise, and strong pairwise monotonic cases in Figure 2. From individual monotonicity to weak pairwise monotonicity to strong pairwise monotonicity, we can obtain a larger geometry. For pairwise monotonicity, the geometry may not be convex.



(a) Individual monotonicity (b) Weak pairwise monotonicity (c) Strong pairwise monotonicity.

Figure 2: Geometry for $\Omega(\mathbf{x})$ with $\mathbf{x}=(3,1)$ induced by individual, weak pairwise, and strong pairwise monotonicity.

5 Fast Marching Method for Optimization

We discuss how to solve equation 11 in this section. The optimization is challenging because of the **nonlinearity** of $\widehat{\mu}(\mathbf{x})$ and $\widehat{\sigma}(\mathbf{x})$, **discrete** and continuous features, and potential **non-convex** geometry (from both $\mathbf{x}'\in\Omega(\mathbf{x})$ and constraints $\widehat{\sigma}^2(\mathbf{x}')<\epsilon$). Therefore, standard optimization algorithms may not be sufficient and difficulties of such problems are discussed by Burer & Letchford (2012). Based on the monotonicity results studied in Section 4, we would like to pursue a new approach to find a **global** maximizers using monotonicity. We begin by binning the features, with discussions in Appendix F. After binning, as a convenience, we assume that $x_j\in\mathbb{Z}^+\cup\{0\}$, $j=1,\dots,m$. Using monotonicity, we want to go through the monotonic sequence

$$\begin{cases}\mathbf{x}^1\to\mathbf{x}^2\to\dots,\\ \text{where } \widehat{\mu}(\mathbf{x}^i)\geq\widehat{\mu}(\mathbf{x}^{i+1}).\end{cases} \tag{19}$$

and we stop when $\widehat{\sigma}^2(\mathbf{x}^i)<\epsilon$. By brute-force calculation, all possible points in the space must be calculated and sorted, which can be very expensive. Therefore, we intend to carry out this process iteratively. For each point $\mathbf{x}$, we want to check its neighbor points in the domain $\Omega(\mathbf{x})$ that have not yet been examined. Define $\mathbf{e}_i$ as

$$(\mathbf{e}_i)_j=\begin{cases}1, & \text{if } j=i,\\ 0, & \text{otherwise.}\end{cases}$$

Each time we iterate, we explore $\psi(\mathbf{x},\mathbf{e}_i)$ for all i, that is, we consider decreasing one unit of the feature. Due to Theorem 4.2, we only need to include maximum boundary points by pairwise monotonicity in the initial set defined recursively as

$$l(\mathbf{x}) = \mathbf{x} \cup \varphi(\mathbf{x}, \mathbf{p}) \cup \bigcup_{i: x_{u_i} > x_{v_i}} l(\mathbf{x}' | \mathbf{x}' = \Gamma(\mathbf{x}, u_i, v_i)). \tag{20}$$

In each iteration, we define our search as follows:

$$\phi(\mathbf{x}) = \bigcup_{i: x_i > 0} \psi(\mathbf{x}, \mathbf{e}_i). \tag{21}$$

As a result, we develop the marching method for optimization (MMO).

Algorithm 1 (Fast) Marching Method for Optimization ((F)MMO)

```
**Inputs**: $\mathbf{x}$, $\widehat{\mu}(\mathbf{x})$, $\widehat{\sigma}(\mathbf{x})$, and a set $l$ defined in equation 20
**Outputs**: Return the **global** optimizer to equation 11 if exists
**while** $\widehat{\sigma}^2(\mathbf{x}) \geq \epsilon$ and $|l| > 0$ **do**
    $l = l \cup \{\mathbf{x}' | \mathbf{x}' \in \phi(\mathbf{x})$ and $\mathbf{x}'$ has not been visited$\}$
    Return $\mathbf{x}$ as the element corresponds to maximum $\widehat{\mu}(\mathbf{x}')$ in $l$ and remove it from $l$ (by Heap)
**end while**
```

The most expensive calculation in the marching algorithm is to determine the maximum value in the set l. A straightforward calculation costs $\mathcal{O}(|l|)$. The heap data structure can accelerate such calculations. Marching with the acceleration by heap is referred to as the Fast Marching Method for Optimization (FMMO). As there are more insertions than extract-max operations, we use the Fibonacci heap by Fredman & Tarjan (1987), which has a lower insertion cost than the binary heap. The marching process with the acceleration by heap has been proven to be a highly effective method for tracing interface evolution by solving the Eikonal equation (Tsitsiklis, 1995; Sethian, 1996; Helmsen et al., 1996). To distinguish our method, we refer to it as the Fast Marching Method for the Eikonal equation (FMME). A discussion of the comparison between FMMO and FMME can be found in Section 7.

5.1 Analysis of the Algorithm

The algorithm is now briefly analyzed with the proof left in Appendix B. The following proposition shows that the algorithm marches monotonically.

Proposition 5.1. *MMO searches for the solution in a monotonic non-increasing order of $\widehat{\mu}$.*

To ensure that we do not miss any points during the march, we provide the following proposition.

Proposition 5.2. *When MMO runs to the end with $l = \{\}$, all points in the domain have been explored.*

Let us suppose that the iteration stops after N steps. Then the space complexity is $\mathcal{O}(mN)$. It is estimated that the amortized time for inserting is $\Theta(1)$ and deleting the maximum key is $\Theta(\log(|l|))$. We further assume that the calculation of $\widehat{\mu}(\mathbf{x})$ and $\widehat{\sigma}(\mathbf{x})$ is C. As a result, at each iteration, the amortized time is $\Theta(Cm + \log(|l|))$. The overall cost of FMMO is

$$\text{Cost}_{FMMO} = \Theta((Cm + \log(m))N + N\log(N)).$$

As a comparison, the cost of MMO is

$$\text{Cost}_{MMO} = \mathcal{O}(CmN + mN^2).$$

As a result, the operations of finding the maximum values will be very expensive for large N.

Remark 5.3. *In practice, the most expensive part is CmN, where C depends on the architectures of models. The number of iterations is therefore the most important factor in computing the cost.*

6 Empirical Example

Experiments are done in high-stakes sectors, including finance, criminology, education, and healthcare. The results are summarized in Table 1. In all experiments, the monotonic groves of neural additive models (MGNAMs) proposed by Chen & Ye (2023) are used. An overview of MGNAMs is provided in Appendix C. For all examples, we use ten models ($M = 10$) for ensembles and threshold $\epsilon = 10^{-3}$. Note models and thresholds are not unique choices and we provide more discussions in Appendix F. We provide detailed analyses for two datasets and leave the remaining examples to the Appendix D. For the baseline methods for solving MINLP, we use Couenne (Belotti et al., 2009). We set all parameters by default. The detailed information on this package is provided in Appendix E.2.

In stage I, we identify the unconfident set $\mathbb{S}$ as in equation 8. In stage II, we solve for optimization equation 11. If for the global optimizer $\mathbf{x}$, $\widehat{\mu}(\mathbf{x}) > \tau$, then they are left in the undecided set $\mathbb{V}(\tau)$, as in equation 12. Our analysis takes into account a variety of choices of τ, which users can select according to their risk appetites. It is important to emphasize that while $\tau = 50\%$ is a natural choice of classification applications, it is not necessarily the best choice for other applications, such as credit scoring. Credit scoring aims to predict the probability of default accurately, and 50% is already a very high probability.

Accuracy metric. For accuracy, we calculate $\frac{|\mathbb{V}(\tau)|}{|\mathbb{S}|} \in [0, 1]$, which is used to determine the ratio of unconfident samples that cannot be provided with reliable lower bounds. Thus, 0 suggests that confident lower bounds are provided for all unconfident samples, whereas 1 suggests that no confident lower bounds are provided.

Efficiency metric. As discussed in Remark 5.3, the iteration number is the key factor in checking efficiency. We calculate average iteration numbers for unconfident samples with $\{\mathbf{x}|\widehat{\mu}(\mathbf{x}) \geq \tau \text{ and } \widehat{\sigma}^2(\mathbf{x}) \geq \epsilon\}$ for different τ.

Table 1: Performance comparison across datasets with various τ values. Better results are bolded.

DATASETS	**τ**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**	**0**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**	**0**
	METHOD	$\frac{\lvert\mathbb{V}(\tau)\rvert}{\lvert\mathbb{S}\rvert}\%$						**AVERAGE ITERATIONS**					
GMSC	FMMO	**77.4**	**64.5**	**43.2**	**15.7**	**7.9**	**1.0**	**24**	**23**	**23**	**25**	**26**	**29**
	Baseline	81.1	70.6	50.7	24.4	15.6	8.1	224	244	254	249	246	253
COMPAS	FMMO	**97.7**	**86.9**	**80.1**	**74.0**	**72.5**	**72.2**	**4**	**3**	**4**	**4**	**4**	**4**
	Baseline	98.0	87.3	80.5	74.4	72.9	72.6	24	17	21	21	21	21
Law School	FMMO	**62.6**	**44.2**	**27.1**	**11.5**	**6.7**	**6.7**	**65**	108	201	501	631	631
	Baseline	66.3	48.3	31.5	18.5	13.1	13.1	88	**96**	**92**	**92**	**94**	**94**
Mammography	FMMO	77.8	76.7	76.4	**74.1**	**71.7**	70.7	**4**	**4**	**4**	**5**	12	12
	Baseline	77.8	76.7	76.4	76.0	74.1	70.7	6	6	6	6	**6**	**6**

6.1 Finance - Credit Scoring

We use the Kaggle credit score dataset, Give Me Some Credit (GSMC). WLOG, we let $x_1 - x_3$ denote the number of past dues and their duration: 90+ days, 60-89 days, and 30-59 days. By domain knowledge, the probability of default is strongly monotonic w.r.t x_1 over x_2 over x_3. Furthermore, we impose individual monotonicity for monthly income (x_4) and number of dependents (x_5).

6.1.1 Stage I - Detect OOD Data

As the first step, we apply the ensemble method to detect OOD data. Running the experiment with the entire dataset leads to the identification of approximately 2.8% of the data as uncertain samples, which are therefore categorized in the unconfident set $\mathbb{S}$. A common example would be an applicant with a high amount of past dues, which is very rare in the dataset. Considering the rarity of this prediction, it makes sense that

Table 2: Examples of GMSC

	Successful Example							Unsuccessful Example						
ITER	x_1	x_2	x_3	x_4	x_5	$\hat{\mu}$	$\hat{\sigma}^2$	x_1	x_2	x_3	x_4	x_5	$\hat{\mu}$	$\hat{\sigma}^2$
0	3	1	3	1	2	0.61	0.0017	0	0	1	1	2	0.42	0.0038
1	3	0	4	1	2	0.60	0.0016	0	0	1	1	1	0.40	0.0037
2	3	1	2	1	2	0.60	0.0015	0	0	1	1	0	0.38	0.0033
3	3	0	3	1	2	0.59	0.0014	0	0	1	0	2	0.38	0.0043
4	3	1	3	1	1	0.59	0.0017	0	0	1	0	1	0.36	0.0041
5	2	2	3	1	2	0.59	0.0010	0	0	1	0	0	0.34	0.0036
6	3	0	4	1	1	0.59	0.0017	0	0	0	1	2	0.23	0.0020
7	3	1	1	1	2	0.59	0.0014	0	0	0	1	1	0.22	0.0018
8	3	1	2	1	1	0.58	0.0016	0	0	0	1	0	0.21	0.0015
9	2	1	4	1	2	0.58	0.0010	0	0	0	0	2	0.20	0.0019
10	2	2	2	1	2	0.58	0.0009	0	0	0	0	1	0.19	0.0017
11								0	0	0	0	0	0.18	0.0014

our model is unconfident about it. The existence of OOD data seems to be commonplace. Consequently, we should exercise caution when applying our models to new situations.

6.1.2 Stage II - Finding Lower Bounds

As a result of considering lower τ, we are able to determine more confident predictions, as expected. By FMMO, only 0.22% of the entire dataset is undecided when $\tau = 0.1$.

A successful example. To demonstrate how FMMO performs, we provide a successful example with

$$\mathbf{x} = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & 0.96 & 0.38 & 9 & 1 & 40 \end{bmatrix}. \tag{22}$$

The iterations of FMMO are recorded in Table 2. By following a monotonic sequence, variance has been reduced to meet the threshold. The reason for this result is that a large number of past dues are rare. The overall number of past dues $(x_1 + x_2 + x_3)$ is greater than 7 in only 232 samples. As the number of past dues decreases, the model becomes more confident in its prediction. Additionally, strong pairwise monotonicity is necessary to go from $(x_1, x_2, x_3) = (3, 1, 3)$ to $(x_1, x_2, x_3) = (2, 2, 2)$.

An unsuccessful example. Unfortunately, not all cases can result in positive outcomes. In some cases, it may not be possible to reduce the variance to the desired level. As in equation 11, we might have

$$\min_{\mathbf{x}' \in \Omega(\mathbf{x})} \widehat{\sigma}^2(\mathbf{x}') \geq \epsilon.$$

The main reason for this is that the uncertainty is primarily derived from nonmonotonic features. More specifically, suppose we split $\mathbf{x}$ into monotonic and nonmonotonic parts as $\mathbf{x} = (\mathbf{x}_{\boldsymbol{\alpha}}, \mathbf{x}_{\neg})$, if $\mathbf{x}_{\neg}$ is the main reason that $\mathbf{x}$ is being OOD, then the maximize may not exist in equation 11. Here is an example,

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 0.90 & 1.06 & 25 & 10 & 57 \end{bmatrix}. \tag{23}$$

The iterations of FMMO are recorded in Table 2. In spite of the fact that FMMO has been run to the end and the variance has been significantly reduced, we are still unable to find the global maximizer for equation 11. In this example, it appears that x_9 is a large value that is substantially different from the mean value of 1. In fact, there are only 113 samples with $x_9 \geq 10$, which suggests that this feature is quite rare and may result in a high level of variance.

6.2 Criminology - Recidivism

We present another example with a less satisfactory result and analyze the reason and potential remedy. In criminology, we examine the prediction of recidivism using the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) (Pro, 2016). There are four monotonic features. Although we can obtain reliable bounds for some data, the performance is not as good as that of the GMSC dataset.

Reasons for the less satisfactory result. There is a difficulty encountered when using the FMMO to reduce the variance of OOD data. For example, when $\tau = 0.3$, there are only 20% points in the unconfident set that find their global maximizers. Age appears to be a significant factor, as demonstrated in Appendix D, but there is no indication that it is a global monotonic feature. The relationship between age and the prediction variance, however, shows a clear pattern, as in Figure 3. Roughly speaking, model predictions are much less confident for young people. We expect the performance to improve significantly if we can incorporate further domain knowledge regarding age, such as local monotonicity for example. Even so, because we are not experts in the field, we do not impose further limitations on age here to avoid unfair treatment.

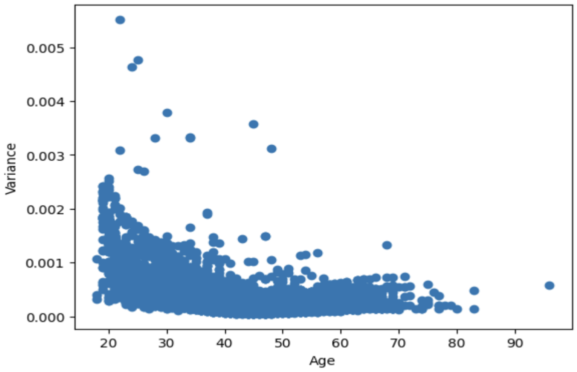


Figure 3: Variance vs Age in the COMPAS dataset.

6.3 Comparison to the Baseline Method

The results of the present study are compared with those of Chen (2022). The stage I is conducted the same for both methods. In stage II of Chen (2022)'s work, only individual monotonicity is assumed. If $\boldsymbol{\alpha}$ contains the index of all monotonic features such that $\mathbf{x} = (\mathbf{x}_{\boldsymbol{\alpha}}, \mathbf{x}_{\neg})$, then equation 11 reduces to

$$\begin{cases} \max_{\mathbf{x}' \in \Omega(\mathbf{x})} \widehat{\mu}(\mathbf{x}'), \\ \text{subject to } \widehat{\sigma}^2(\mathbf{x}') < \epsilon, \\ \Omega(\mathbf{x}) = \{\mathbf{x}' | \mathbf{x}'_{\boldsymbol{\alpha}} \leq \mathbf{x}_{\boldsymbol{\alpha}}\}. \end{cases} \tag{24}$$

Now that the geometry has been explicitly specified, we follow Chen (2022) and use the existing state-of-the-art MINLP package Couenne (Belotti et al. 2009)[3]. More details are provided in Appendix E. In summary, we have made two major improvements over Chen & Ye (2022).

- In Chen (2022), only individual monotonicity is considered in the optimization and it solves for equation 24. Using general monotonicity, we solve the problem for equation 11, allowing a larger domain $\Omega(\mathbf{x})$, thus improving accuracy.
- Equation 11 presents a complicated MINLP that the global optimizer can't guarantee to solve efficiently. Chen (2022) relies on the existing optimization package that aims for general MINLP. Our proposed FMMO has improved the accuracy of the problem by incorporating monotonicity to efficiently find the **global** optimizer to equation 11.

A summary of the results is presented in Table 1. Our method has consistently improved accuracy since it determines the global maximizer and pairwise monotonicity is included. Overall, we have observed that our method has improved accuracy greatly, with the exception of the Mammography dataset. In the case

[3] https://github.com/coin-or/Couenne

of the Mammography dataset, we find that the model is generally not confident about its prediction (the percentage of OOD data is greater than 50%). Thus, most unconfident samples do not have good bounds. Secondly, iteration numbers are not necessarily more than the baseline method, except for the law school dataset. On the law school dataset, we observe that it is quite difficult for the FMMO for some data points to find the global optimizer and run the algorithm to the end, thus resulting in a high number of iterations.

7 Related Work

Relationship to Monotonic Models There has been considerable interest in monotonic models in the ML community, including Runje & Shankaranarayana (2023); You et al. (2017); Milani Fard et al. (2016); Daniels & Velikova (2010); Sill (1997); Liu et al. (2020); Bakst et al. (2020); Sivaraman et al. (2020); Sill & Abu-Mostafa (1996). More details are provided in Appendix G.1. While our methods are based on monotonic models as in equation 11, the focus of this research is not on how monotonic models are trained but rather on how monotonicity can be further utilized to reduce predictive uncertainty. In this way, our method can be applied to general monotonic models to reduce their predictive uncertainty.

Relationship to Out-of-distribution Detection There have been studies of OOD detection methods that have been successful and detailed comparisons of these methods can be found in Ovadia et al. (2019); Yang et al. (2022). More details are provided in Appendix G.2. In traditional OOD tasks, the main focus is on detecting OOD data. As a result, the system could abstain from making decisions due to low confidence(Ovadia et al., 2019; OCC, 2021). We focus on the case of underlying models that have prior knowledge of monotonicity, and how a confident alternative point can be derived by assuming monotonicity to provide additional information of unconfident predictions. Therefore, the choice of OOD methods may not be unique. As part of this work, we examine ensemble methods and use their variance as a proxy for uncertainty, but it is possible to extend this approach by considering other measures of uncertainty. A simple way to accomplish this is to switch $\widehat{\sigma}^2(\mathbf{x}) < \epsilon$ to other constraints in equation 11.

Relationship to Fast Marching Methods for Solving the Eikonal Equation Originally, the FMME (Tsitsiklis, 1995; Sethian, 1996; Helmsen et al., 1996) was proposed as a solution to the Eikonal equation in numerical partial differential equations. More details of FMME are provided in Appendix G.3. FMMO and FMME address very different problems, but both utilize a marching concept that is accelerated by heap data structures.

8 Limitations and Future Work

There are some limitations of our current approach, as summarized below. Correspondingly, we propose future research.

1. The major limitation of the current FMMO is that it does not consider the behavior of prediction variance, thus it may take a considerable number of iterations, especially for high-dimensional problems. We plan to utilize properties of model variances to improve the FMMO's search process.
2. In general, FMMO performs better when more features exhibit monotonicity, especially important features. It should be noted, however, that some important features may not exhibit monotonicity, at least not globally. The performance may be improved by applying other domain knowledge (Gupta et al., 2020; 2018) and imposing local monotonicity. Such a direction will be explored.

References

Propublica. compas data and analysis for “machine bias"., 2016. URL https://github.com/propublica/compas-analysis.

Model risk management, 2021. URL https://www.occ.treas.gov/publications-and-resources/publications/comptrollers-handbook/files/model-risk-management/index-model-risk-management.html.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34, 2021.

William Taylor Bakst, Nobuyuki Morioka, and Erez Louidor. Monotonic kronecker-factored lattice. In *International Conference on Learning Representations*, 2020.

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009.

Koby Bibas, Meir Feder, and Tal Hassner. Single layer predictive normalized maximum likelihood for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:1179–1191, 2021.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Samuel Burer and Adam N Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

Saul Calderon-Ramirez, Diego Murillo-Hernandez, Kevin Rojas-Salazar, Luis-Alexander Calvo-Valverd, Shengxiang Yang, Armaghan Moemeni, David Elizondo, Ezequiel López-Rubio, and Miguel A Molina-Cabello. Improving uncertainty estimations for mammogram classification using semi-supervised learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

José-Ramón Cano, Pedro Antonio Gutiérrez, Bartosz Krawczyk, Michał Woźniak, and Salvador García. Monotonic classification: An overview on algorithms, performance measures and data sets. *Neurocomputing*, 341:168–182, 2019.

Dangxing Chen. Two-stage modeling for prediction with confidence. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1–5. IEEE, 2022.

Dangxing Chen and Weicheng Ye. Monotonic neural additive models: Pursuing regulated machine learning models for credit scoring. In *Proceedings of the Third ACM International Conference on AI in Finance*, pp. 70–78, 2022.

Dangxing Chen and Weicheng Ye. How to address monotonicity for model risk management? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5282–5295. PMLR, 23–29 Jul 2023.

Andrew Cotter, Maya Gupta, Heinrich Jiang, Erez Louidor, James Muller, Tamann Narayan, Serena Wang, and Tao Zhu. Shape constraints for set functions. In *International conference on machine learning*, pp. 1388–1396. PMLR, 2019.

Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6):906–917, 2010.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Matthias Elter. Mammographic Mass. UCI Machine Learning Repository, 2007. DOI: https://doi.org/10.24432/C53K6Z.

Michael L Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

Eric Friedman and Herve Moulin. Three methods to share joint costs or surplus. *Journal of economic Theory*, 87(2):275–312, 1999.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Maya Gupta, Dara Bahri, Andrew Cotter, and Kevin Canini. Diminishing returns shape constraints for interpretability and regularization. *Advances in neural information processing systems*, 31, 2018.

Maya Gupta, Erez Louidor, Oleksandr Mangylov, Nobu Morioka, Taman Narayan, and Sen Zhao. Multidimensional shape constraints. In *International Conference on Machine Learning*, pp. 3918–3928. PMLR, 2020.

William E Hart, Jean-Paul Watson, and David L Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3:219–260, 2011.

John Joseph Helmsen, Elbridge Gerry Puckett, Phillip Colella, and Milo Dorr. Two new methods for simulating photolithography development in 3d. In *Optical Microlithography IX*, volume 2726, pp. 253–261. SPIE, 1996.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Navid Kardan, Ankit Sharma, and Kenneth O Stanley. Towards consistent predictive confidence through fitted ensembles. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Xingchao Liu, Xing Han, Na Zhang, and Qiang Liu. Certified monotonic neural networks. *Advances in Neural Information Processing Systems*, 33:15427–15438, 2020.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pp. 2218–2227. PMLR, 2017.

Daniel D Lundstrom, Tianjian Huang, and Meisam Razaviyayn. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pp. 14485–14508. PMLR, 2022.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082, 15406261.

Mahdi Milani Fard, Kevin Canini, Andrew Cotter, Jan Pfeifer, and Maya Gupta. Fast and flexible monotonic functions with ensembles of lattices. *Advances in neural information processing systems*, 29, 2016.

Shayan Shaghayeq Nazari and Pinku Mukherjee. An overview of mammographic density and its association with breast cancer. *Breast cancer*, 25:259–267, 2018.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Arlie O Petters and Xiaoying Dong. An introduction to mathematical finance with applications. *New York, NY: Springer*, 10:978–1, 2016.

Marco Repetto. Multicriteria interpretability driven deep learning. *Annals of Operations Research*, pp. 1–15, 2022.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

Davor Runje and Sharath M Shankaranarayana. Constrained monotonic neural networks. In *International Conference on Machine Learning*, pp. 29338–29353. PMLR, 2023.

James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

Lloyd S Shapley et al. A value for n-person games. 1953.

Joseph Sill. Monotonic networks. *Advances in neural information processing systems*, 10, 1997.

Joseph Sill and Yaser Abu-Mostafa. Monotonicity hints. *Advances in neural information processing systems*, 9, 1996.

Aishwarya Sivaraman, Golnoosh Farnadi, Todd Millstein, and Guy Van den Broeck. Counterexample-guided learning of monotonic neural networks. *Advances in Neural Information Processing Systems*, 33:11936–11948, 2020.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *Advances in neural information processing systems*, 28, 2015.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269–9278. PMLR, 2020.

John N Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE transactions on Automatic Control*, 40(9):1528–1538, 1995.

Serena Wang and Maya Gupta. Deontological ethics by monotonicity shape constraints. In *International conference on artificial intelligence and statistics*, pp. 2043–2054. PMLR, 2020.

Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, WENXUAN PENG, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. OpenOOD: Benchmarking generalized out-of-distribution detection. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. *Advances in neural information processing systems*, 30, 2017.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

A Geometry of the Domain

It is sufficient to examine different monotonicity separately, as pairwise monotonic features are assumed to be disjoint. Proofs are in Appendix B.

A.1 Individual Monotonicity

In the case that $x_1, \ldots, x_m$ only exhibit individual monotonicity, the geometry resulting from the individual monotonicity is a box.

Proposition A.1. *Suppose f is individually monotonic with respect to $x_1, \ldots, x_m$, then*

$$\Omega(\mathbf{x}) = \{\mathbf{x}' | \mathbf{x}' \leq \mathbf{x}\} = \{(x_1', \ldots, x_m') | x_1' \leq x_1, \ldots, x_m' \leq x_m\}.$$

A.2 Weak Pairwise Monotonicity

Pairwise monotonicity presents a more challenging situation. Firstly, we will identify the maximum boundary points, followed by determining the interior points. By identifying the maximum boundary points, we will be able to determine whether a point belongs to the desired geometry.

With the size of $x_\beta + x_\gamma$ fixed, we obtain the following proposition to identify whether we are able to make the comparison under weak pairwise monotonicity.

Proposition A.2. *Suppose f is weakly pairwise monotonic with respect to x_β over x_γ, then $f(\Gamma(\mathbf{x}, \beta, \gamma)) \underset{M}{\leq} f(\mathbf{x})$, if $x_\beta > x_\gamma$.*

As a result, we can identify the maximum boundary points of the domain. As a next step, we would like to determine the maximum boundary points under weak pairwise monotonicity.

Proposition A.3. *Suppose f is weakly monotonic with respect to x_{u_i} over x_{v_i} for $i = 1, \ldots, |\mathbf{u}|$, then*

$$\partial\Omega(\mathbf{x}) = \bigcup_{i: x_{u_i} > x_{v_i}} \partial\Omega(\mathbf{x}' | \mathbf{x}' = \Gamma(\mathbf{x}, u_i, v_i)). \tag{25}$$

In other words, by fixing the size of $\sum_{i:i\in U} x_i$, we consider all possible swaps. Recursive definitions are used since swaps can be performed more than once. Clearly, for a new point $\mathbf{x}'$, if $\sum_{i=1}^m x_i' > \sum_{i=1}^m x_i$, we don't have enough information to compare it with $\mathbf{x}$. Conversely, we have the following theorem, indicates that all points in the domain are bounded by maximum boundary points.

Theorem A.4. *Suppose f is weakly monotonic with respect to x_{u_i} over x_{v_i} for $i = 1, \ldots, |\mathbf{u}|$, there exists a $\mathbf{x}'$ with $\sum_{i\in U} x_i' < \sum_{i\in U} x_i$ and $f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})$, then there exists a $\widetilde{\mathbf{x}}' \in \partial\Omega(\mathbf{x})$ such that $\mathbf{x}' \leq \widetilde{\mathbf{x}}'$.*

By Theorem A.4, we need only consider the “interior" of $\partial\Omega$ to determine the domain. It should be noted that in this case, the interior has a different definition. We say $\mathbf{x}$ is an interior point of $\partial\Omega$ if $\mathbf{x} \leq \mathbf{x}'$ for some $\mathbf{x}' \in \partial\Omega$. We have the following proposition to provide a formula for the domain under weak pairwise monotonicity, as a result of Proposition A.3 and Theorem A.4.

Proposition A.5. *Suppose f is weakly monotonic with respect to x_{u_i} over x_{v_i}, then*

$$\Omega(\mathbf{x}) = \{\mathbf{x}' | \mathbf{x}' \leq \mathbf{x}\} \cup \bigcup_{i: x_{u_i} > x_{v_i}} \Omega(\mathbf{x}' | \mathbf{x}' = \Gamma(\mathbf{x}, u_i, v_i)). \tag{26}$$

A.3 Strong Pairwise Monotonicity

We then consider the strong pairwise monotonicity. Suppose we have a list $\mathbf{p}$ such that f is strongly pairwise monotonic with respect to x_{p_i} over $x_{p_{i+1}}$ for $i = 1, \ldots, |\mathbf{p}| - 1$. Based on the strong pairwise monotonicity, we can derive the following proposition.

Proposition A.6. *Suppose f is strongly pairwise monotonic with respect to x_β over x_γ, if $x_\beta' \leq x_\beta, x_\gamma' = x_\beta + x_\gamma - x_\beta'$, then $f(x_\beta', x_\gamma', \mathbf{x}_\neg) \underset{M}{\leq} f(x_\beta, x_\gamma, \mathbf{x}_\neg)$.*

Based on this, we obtain the following result.

Theorem A.7. *For $f(x_1, \ldots, x_m)$, where f is strongly pairwise monotonic with respect to x_i over x_{i+1} for $i = 1, \ldots, m-1$, then*

$$\Omega(\mathbf{x}) = \left\{ \mathbf{x}' \middle| x_i' \leq \sum_{j=1}^{i} x_j - \sum_{j=1}^{i-1} x_j', \forall i \right\}. \tag{27}$$

As a result of Theorem A.7, we obtain the following proposition, which allows us to calculate the maximum boundary points.

Proposition A.8. *For $f(x_1, \ldots, x_m)$, where f is strongly pairwise monotonic with respect to x_i over x_{i+1} for $i = 1, \ldots, m-1$, then*

$$\partial\Omega(\mathbf{x}) = \varphi(\mathbf{x}, \mathbf{p}). \tag{28}$$

As a result, if there exists a $\mathbf{x}'$ with $\sum_{i=1}^{m} x'_i < \sum_{i=1}^{m} x_i$ and $f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})$, then there exists a $\widetilde{\mathbf{x}}' \in \partial\Omega(\mathbf{x})$ such that $\mathbf{x}' \leq \widetilde{\mathbf{x}}'$.

B MISSING PROOFS

This section contains detailed proofs of the results that are missing in the main paper.

B.1 Proof of Theorem 3.1

Proof. Suppose $\widehat{f}_i$ is individually monotonic to x_α, then

$$\widehat{f}_i(x_\alpha, \mathbf{x}_\neg) \underset{M}{\leq} \widehat{f}_i(x'_\alpha, \mathbf{x}_\neg), \text{ if } x_\alpha \leq x'_\alpha.$$

If this is true for all i, then for $x'_\alpha \geq x_\alpha$, we have

$$\frac{1}{M}\sum_{i=1}^{M} \widehat{f}_i(x_\alpha, \mathbf{x}_\neg) \underset{M}{\leq} \frac{1}{M}\sum_{i=1}^{M} \widehat{f}_i(x'_\alpha, \mathbf{x}_\neg)$$

Suppose $\widehat{f}_i$ is weakly pairwise monotonic with respect to x_β over x_γ, then for $x_\beta = x_\gamma$, we have

$$\widehat{f}_i(x_\beta, x_\gamma + c, \mathbf{x}_\neg) \underset{M}{\leq} \widehat{f}_i(x_\beta + c, x_\gamma, \mathbf{x}_\neg), \forall c \in \mathbb{R}^+.$$

If this is true for all i, then we have

$$\frac{1}{M}\sum_{i=1}^{M} \widehat{f}_i(x_\beta, x_\gamma + c, \mathbf{x}_\neg) \underset{M}{\leq} \frac{1}{M}\sum_{i=1}^{M} \widehat{f}_i(x_\beta + c, x_\gamma, \mathbf{x}_\neg).$$

A similar conclusion can be drawn for strong pairwise monotonicity. □

B.2 Proof of Proposition A.1

Proof. If $\mathbf{x}' \in \Omega(\mathbf{x})$, then $f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})$ by definition. Conversely, if $x'_i > x_i$ for some i, we cannot draw any conclusions. □

B.3 Proof of Proposition A.2

Proof. Without loss of generality, we write $\mathbf{x} = (x_\beta, x_\gamma, \mathbf{x}_\neg)$. Consider $\mathbf{x}' = (x_\gamma, x_\gamma, \mathbf{x}_\neg)$ and $c = x_\beta - x_\gamma > 0$, then by definition, we have

$$f(x_\gamma, x_\gamma + c, \mathbf{x}_\neg) \underset{M}{\leq} f(x_\gamma + c, x_\gamma, \mathbf{x}_\neg).$$

□

B.4 Proof of Proposition A.3

Proof. From Proposition A.2, we determine the form. Otherwise, if $x_\beta < x_\gamma$, we cannot draw conclusions. □

B.5 Proof of Theorem A.4

Proof. Consider the sequence of steps required to make $\mathbf{x}'$ from $\mathbf{x}$,

$$\mathbf{x}^1 \to \mathbf{x}^2 \to \cdots \to \mathbf{x}',$$

with $\mathbf{x}^1 = \mathbf{x}$. For each step, we can either reduce the value of x_i as long as $x_i' \geq 0$, or swap x_i with x_j if $x_i > x_j$. We would like to construct a new sequence by applying reduction operations and swap operations one by one. In order to accomplish this, we merge all reduction operations between two swap operations. In the absence of a reduction operation, we simply consider $\psi(\mathbf{x}, \mathbf{0})$. As a result, we have

$$\mathbf{x}^{i+1} = \begin{cases} \psi(\mathbf{x}^i, \mathbf{c}^i), & \text{if } i \text{ odd}, \\ \Gamma(\mathbf{x}^i, u^i, v^i), & \text{if } i \text{ even}. \end{cases} \tag{29}$$

Next, we construct another sequence in $\partial\Omega$ to bound $\mathbf{x}^i$,

$$\widetilde{\mathbf{x}}^1 \to \widetilde{\mathbf{x}}^2 \to \cdots \to \widetilde{\mathbf{x}}',$$

with $\widetilde{\mathbf{x}}^1 = \mathbf{x}$ and

$$\widetilde{\mathbf{x}}^{i+1} = \begin{cases} \widetilde{\mathbf{x}}^i, & \text{if } i \text{ odd}, \\ \Gamma(\widetilde{\mathbf{x}}^i, u^i, v^i) & \text{if } i \text{ even and } \widetilde{\mathbf{x}}^i_{u^i} > \widetilde{\mathbf{x}}^i_{v^i}, \\ \widetilde{\mathbf{x}}^i, & \text{if } i \text{ even and } \widetilde{\mathbf{x}}^i_{u^i} < \widetilde{\mathbf{x}}^i_{v^i}. \end{cases}$$

We want to show that $\widetilde{\mathbf{x}}^i \geq \mathbf{x}^i$ for all i. It is clear that this holds for $i = 1$, and we consider when $i > 1$. We focus on the third case because the first two cases are obvious. If $\widetilde{\mathbf{x}}^i_{u^i} < \widetilde{\mathbf{x}}^i_{v^i}$ and $\mathbf{x}^i_{u^i} > \mathbf{x}^i_{v^i}$, since $\mathbf{x}^i_{u^i} \leq \mathbf{x}^{i-1}_{u^{i-1}}$ and $\mathbf{x}^{i-1}_{u^{i-1}} \leq \widetilde{\mathbf{x}}^i_{u^i}$, then $\mathbf{x}^i_{u^i} < \widetilde{\mathbf{x}}^i_{v^i}$ and $\mathbf{x}^i_{v^i} < \widetilde{\mathbf{x}}^i_{u^i}$. Thus, after swapping on $\mathbf{x}^i$, $\mathbf{x}^{i+1} \leq \widetilde{\mathbf{x}}^{i+1}$. By induction, we conclude.

□

B.6 Proof of Proposition A.6

Proof. Let $c = x_\beta - x_\beta'$, then we have

$$f(x_\beta', x_\gamma', \mathbf{x}_\neg) = f(x_\beta', x_\gamma + c, \mathbf{x}_\neg) \underset{M}{\leq} f(x_\beta' + c, x_\gamma, \mathbf{x}_\neg) = f(x_\beta, x_\gamma, \mathbf{x}_\neg).$$

□

B.7 Proof of Theorem A.7

Proof. First, we show if $\mathbf{x}' \in \Omega(\mathbf{x})$, then $f(\mathbf{x}') \underset{M}{\leq} f(\mathbf{x})$. Denote $c_i = x_i - x_i'$, then from Equation equation 27 we have

$$\sum_{j=1}^{i} c_j \geq 0, i = 1, \ldots, m.$$

By Proposition A.6 and individual monotonicity, we have

$$\begin{aligned} f(x_1, \ldots, x_m) &\underset{M}{\geq} f(x_1', x_2 + c_1, \ldots, x_m) \\ &\underset{M}{\geq} f(x_1', x_2', x_3 + c_1 + c_2, \ldots, x_m) \\ &\underset{M}{\geq} \cdots \\ &\underset{M}{\geq} f\left(x_1', \ldots, x_m' + \sum_{i=1}^{m} c_i\right) \\ &\underset{M}{\geq} f(x_1', \ldots, x_m'). \end{aligned}$$

Conversely, suppose $\mathbf{x}' \notin \Omega(\mathbf{x})$, then $\exists i$ such that $\sum_{j=1}^{i} x_j < \sum_{j=1}^{i} x'_j$. Let $c = \sum_{j=1}^{i} x_j$. Consider the function

$$f(x_1, \ldots, x_m) = 1_{\sum_{j=1}^{i} x_j > c}.$$

Clearly, f satisfies the individual and strong pairwise monotonicity. However, we have

$$0 = f(\mathbf{x}) \underset{M}{\leq} f(\mathbf{x}') = 1.$$

Thus, we conclude.

□

B.8 Proof of Proposition 5.1

Proof. By individual monotonicity, we know if $\mathbf{x}' \in \phi(\mathbf{x})$, then $\widehat{\mu}(\mathbf{x}') \underset{M}{\leq} \widehat{\mu}(\mathbf{x})$. □

B.9 Proof of Proposition 5.2

Proof. If there is $\mathbf{x}' \in \Omega(\mathbf{x})$ has not been explored, then $\mathbf{x}' + \mathbf{e}_i$ has not been explored for all i except at boundaries. By Proposition A.3, Theorem A.4 and Proposition A.6, we know $\mathbf{x}' \leq \widetilde{\mathbf{x}}$ for some $\widetilde{\mathbf{x}} \in \partial\Omega(\mathbf{x})$ and all maximum boundary points are included in the initial list. It is possible to reach max boundary points if we continue adding $\mathbf{e}_i$ for some i, as a contradiction. □

C Monotonic Groves of Neural Additive Models

Here, we briefly review the Monotonic Groves of Neural Additive Models (MGNAMs) introduced by Chen & Ye (2023). Assume that the features are divided into individual monotonic, weak pairwise monotonic, strong pairwise monotonic, and nonmonotonic parts, as $\mathbf{x} = (\mathbf{x}_S, \mathbf{x}_U, \mathbf{x}_P, \mathbf{x}_\neg)$, similar to Section 4. For features with weak pairwise monotonicity in U, we give two lists $\mathbf{u}$ and $\mathbf{v}$ with $|\mathbf{u}| = |\mathbf{v}|$ such that f is weakly pairwise monotonic with respect to x_{u_i} over x_{v_i} for $i = 1, \ldots, |\mathbf{u}|$. For strong pairwise monotonicity, we assume that there is a list $\mathbf{p}$ such that f is strongly pairwise monotonic to x_{p_j} over $x_{p_{j+1}}$ for $j = 1, \ldots, |\mathbf{p}| - 1$. All monotonic features follow individual monotonicity. GMNAMs use a special architecture of

$$f(\mathbf{x}; \mathbf{\Theta}) = \alpha + \sum_{s:s\in S} f(x_s; \boldsymbol{\theta}_s) + \sum_{u:u\in U} f(x_u; \boldsymbol{\theta}_u) + f(\mathbf{x}_P; \boldsymbol{\theta}_P) + \sum_{\gamma\in\neg} f(x_\gamma, \boldsymbol{\theta}_\gamma),$$

where each f is parametrized by neural networks, $\boldsymbol{\theta}_s, \boldsymbol{\theta}_u, \boldsymbol{\theta}_P, \boldsymbol{\theta}_\gamma$ are parameters for each individual neural networks, and $\mathbf{\Theta}$ include all parameters. The architecture is assumed to keep the model transparent, where $f(x, \boldsymbol{\theta}_s), f(x, \boldsymbol{\theta}_u), f(x, \boldsymbol{\theta}_\gamma)$ only takes one feature and $f(\mathbf{x}, \boldsymbol{\theta}_P)$ is a function of several features with strong pairwise monotonicity. This architecture is assumed since if the function is additively separated, then strong pairwise monotonic features have difficulty exhibiting diminishing marginal effects, which is common in social science. Following this, the standard regularization methods are applied

$$\min_{\mathbf{\Theta}} \ell(\mathbf{\Theta}) + \lambda_1 h_1(\mathbf{\Theta}) + \lambda_2 h_2(\mathbf{\Theta}) + \lambda_3 h_3(\mathbf{\Theta}), \tag{30}$$

where $\ell(\Theta)$ is the mean-squared error for regressions and log-likelihood function for classification, and h_1, h_2, h_3 are corresponding penalties for individual, weak pairwise, and strong pairwise monotonicity. Let us assume that all features are already binned with sets $\mathbb{S}_i$ for each $s \in S$, $\mathbb{U}_i$ for each $u \in U$, and $\mathbb{P}$. To simplify the notation, we assume that all features are binned equally with the distance Δx. As a result, we

Table 3: Summary results for all datasets

DATASETS	GMSC	COMPAS	LAW	LIFE-SCIENCE	MAMMOGRAPHY
AUC (%)	85	72	86	68	90
OOD (%)	2.8	10.4	10.5	14.0	52.2

have

$$h_1(\mathbf{\Theta}) = \sum_{s\in S}\sum_{x_i\in\mathbb{S}_i} \max\left(0, f(x_i+\Delta x;\boldsymbol{\theta}_s) - f(x_i;\boldsymbol{\theta}_s)\right),$$

$$h_2(\mathbf{\Theta}) = \sum_{u\in U}\sum_{x_i\in\mathbb{U}_i} \max\left(0, f(x_i+\Delta x;\boldsymbol{\theta}_u) - f(x_i;\boldsymbol{\theta}_u)\right) + \sum_{j=1}^{|\mathbf{u}|}\sum_{x_i\in\mathbb{U}_i} \max(0, f(x_i;\boldsymbol{\theta}_{u_j}) - f(x_i;\boldsymbol{\theta}_{v_j})),$$

$$h_3(\mathbf{\Theta}) = \sum_{p\in P}\sum_{\mathbf{x}_i\in\mathbb{P}} \max(0, f(\mathbf{x}_i+\mathbf{e}_p;\boldsymbol{\theta}_P) - f(\mathbf{x}_i;\boldsymbol{\theta}_P)) + \sum_{j=1}^{|\mathbf{p}|-1}\sum_{\mathbf{x}_i\in\mathbb{P}} \max(0, f(\mathbf{x}_i+\mathbf{e}_{p_j};\boldsymbol{\theta}_P) - f(\mathbf{x}_i+\mathbf{e}_{p_{j+1}};\boldsymbol{\theta}_P)),$$

whereas $(\mathbf{e}_i)_j = \delta_{i,j}$, whereas $\delta_{i,j}$ is the kronecker delta. The model is trained by algorithm 2 .

Algorithm 2 Monotonic Groves of Neural Additive Models (MGNAMs)

```
**Initialization**: $\lambda_1 = \lambda_2 = \lambda_3 = 0$, the architecture of the groves of neural additive models $(S, U, P, \neg)$
Train a groves of neural additive model by equation 30
**while** $\min(h_1, h_2, h_3) > 0$ **do**
    Increase $\lambda_i$ if $h_i > 0$
    Update the groves of neural additive models by equation 30.
**end while**
```

D DATA and MODELS

A summary of the results is presented in Table 3. Accuracy of models is evaluated by the AUC. We find that the AUC is almost the same before and after monotonicity is imposed, which is consistent with the findings of Chen & Ye (2023); Wang & Gupta (2020); Gupta et al. (2020). As a reminder, monotonicity is primarily intended for the purposes of conceptual soundness and fairness. Further, this paper does not focus on the accuracy of the model, but rather on the effectiveness of finding global maximums by equation 11. Reporting the results of accuracy is only for the purpose of completeness.

D.1 Finance - Credit Scoring

D.1.1 Data Description

We use the Kaggle credit score dataset [4].

- $x_1 - x_3$: Last two years, the number of times borrower was 90+ days past due, 60-89 days past due, and 30-59 days past due.
- x_4: Monthly income.
- x_5: Number of dependents in the family.
- x_6: Total balance on credit cards and personal lines of credit except for real estate and no installment debt such as car loans divided by the sum of credit limits.

[4] https://www.kaggle.com/c/GiveMeSomeCredit/overview

- x_7: Monthly debt payments, alimony, and living costs divided by monthly gross income.
- x_8: Number of open loans and lines of credit
- x_9: Number of mortgage and real estate loans
- x_{10}: Age of borrower in years.
- y: Client's behavior; 1 = Person experienced 90 days past due delinquency or worse.

We impose strong pairwise monotonicity of $x_1 - x_3$ and individual monotonicity for $x_4 - x_5$.

For simplicity, data with missing variables are removed. Past dues that are greater or equal to 20 are discarded. Then past dues greater than four times are replaced by four due to the rarity. This also applies to x_5 if its value exceeds five. To apply the fast marching method, we categorize x_4 into the following intervals: $[0, \$2500)$, $[\$2,500, \$5,000)$, $[\$5,000, \$7,500)$, $[\$7,500, \$10,000)$, $[\$10,000, \$50,000)$, and $[\$50,000, \infty)$. Afterward, they are transformed from five to zero so that f increases monotonically with respect to x_4. We make such a choice in order to make features as easy to understand as possible for customers. This is not a unique choice. The model performance has been monitored to ensure that the accuracy does not deteriorate. When checking for accuracy, the dataset is randomly partitioned into 70% training and 30% test sets.

D.1.2 Model

For MGNAM, we consider the architecture

$$f(\mathbf{x}) = f_{1,2,3}(x_1, x_2, x_3) + f_4(x_4) + \cdots + f_{10}(x_{10}). \tag{31}$$

In other words, $x_1 - x_3$ are grouped together, and the remaining features are handled using 1-dimensional functions. For $x_1 - x_3$, we enforce strong pairwise monotonicity. We enforce individual monotonicity for $x_4 - x_5$. All functions are approximated by neural networks with one hidden layer of four neurons. We focus on simple architectures since there is no apparent improvement in accuracy for more complicated models.

D.1.3 Results

The area-under-the-curve (AUC) of the model is around 85%, which indicates that the model is accurate. It might be possible to improve model performance by further cleaning the data, but since this is not the primary concern of our study, we opt to omit it for simplicity.

D.2 Criminology - Recidivism

D.2.1 Data Description

COMPAS is a proprietary score developed to predict recidivism risk, which is used to guide bail, sentencing, and parole decisions. A report published by ProPublica in 2016 provided recidivism data for defendants in Broward County, Florida (Pro, 2016). We focus on the simplified cleaned dataset provided in Dressel & Farid (2018). Three thousand and fifty-one (45%) of the 7,214 observations committed a crime within two years. This study uses a binary response variable, recidivism, as the response variable. The dataset here contains nine features selected after some feature selection was conducted.

- x_1: Total number of juvenile felony criminal charges
- x_2: Total number of juvenile misdemeanor criminal charges
- x_3: Age
- x_4: Total number of non-juvenile criminal charges
- x_5: A numeric value corresponding to the specific criminal charge

- x_6: An indicator of the degree of the charge: misdemeanor or felony
- x_7: Races include White (Caucasian), Black (African American), Hispanic, Asian, Native American, and Others
- x_8: Sex, male or female
- x_9: A numeric value between 1 and 10 corresponds to the recidivism risk score generated by COMPAS software (a small number corresponds to a low risk, and a larger number corresponds to a high risk)
- y: Whether the defendant recidivated two years after the previous charge

To avoid discrimination, we further exclude races and sexes. The COMPAS score is also excluded as it is not the focus of this study and is correlated with other features, making its interpretation more difficult. As there are too few samples, we truncate the number of juveniles exceeding five. Otherwise, if monotonicity is requested, neural network functions will become flat, which is not helpful.

D.2.2 Model

For MGNAM, we consider the architecture

$$f(\mathbf{x}) = f_{1,2}(x_1, x_2) + f_3(x_3) + \cdots + f_6(x_6). \tag{32}$$

In other words, $x_1 - x_2$ are grouped, and the remaining features are handled using 1-dimensional functions. For $x_1 - x_2$, we enforce strong pairwise monotonicity. We further impose individual monotonicity on x_4 and x_6.

D.2.3 Result

The AUC of the model is about 72%, which is consistent with the literature (Dressel & Farid, 2018).

We calculate the global feature importance by BShap in Figure 4. A brief description of BShap is provided in Section H. We take the mean value as the baseline value $\mathbf{x}'$. This result indicates that x_3, the Age, is an essential feature.

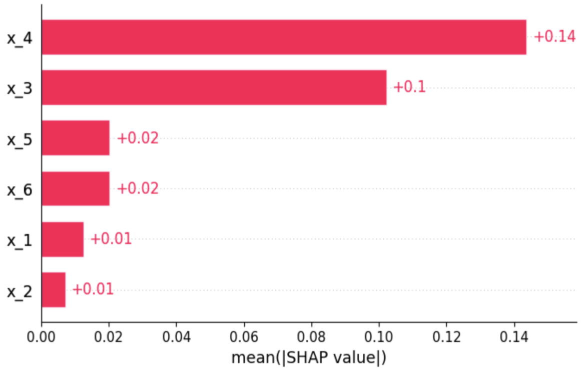


Figure 4: Global feature importance of COMPAS using BShap.

D.3 Education - Law School Bar Exam

D.3.1 Data description

The law school dataset (Wightman, 1998) concerns information on the probability of passing the bar examination. In 1991, 163 law schools in the United States were surveyed by the Law School Admission Council (LSAC). From the total of 18,692 observations, 16,856 (90%) people passed the bar for the first time. If, for instance, universities wish to award scholarships based on the likelihood of passing the bar examination, fairness could be important. In this study, the response variable is a binary variable, pass. There are 11 features in this dataset.

- x_1: The student's decile in the school given his grades in Year 3
- x_2: The student's decile in the school given his grades in Year 1
- x_3: The student's LSAT score
- x_4: The student's undergraduate GPA
- x_5: Whether the student will work full-time or part-time
- x_6: The student's family income bracket
- x_7: Tier, which is an indicator of school quality
- x_8: Whether the student is a male or female
- x_9: Race
- x_{10}: The student's first-year law school GPA
- x_{11}: The student's cumulative law school GPA
- y: Whether the student passed the bar exam on the first try

Race and sex were excluded for potential bias. The law school GPA (LGPA) is calculated on different scales for the first year and the cumulative. To make a comparison, we scale them. $x_{10} - x_{11}$ are excluded as they are highly correlated with $x_1 - x_2$. Additionally, to avoid unfairness, gender and race are also excluded. Hence, the first 7 features remained to train the model.

D.3.2 Model

For MGNAM, we consider the architecture

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_7(x_7). \tag{33}$$

For all grade-related features ($x_1 - x_4$), we require individual monotonicity, as well as weak pairwise monotonicity for x_1 over x_2. In the latter cases, the requirement indicates the pairwise monotonicity of time: the more recent information should be regarded as more valuable.

D.3.3 Results

The AUC of the model is about 86%. Regarding the FMMO results, approximately 86 percent of OOD data obtained a confident lower bound. The global BShap value is calculated in Figure 5. There is great significance to the x_1, x_2, and x_7 features. This model is designed to ensure fairness by not considering x_7, the tier of the law school, as a monotonic feature. However, the tier is an important feature that contributes to the uncertainty associated with the prediction. Taking the example of Figure 6, high variance data can be observed in cases where the law school's tier is high and the student's LSAT is low. Since most admitted students to top-tier schools possess high LSAT scores, this is intuitively reasonable. Without the monotonic information of tier, high-variance data cannot be effectively handled. To mitigate bias, monotonicity on tier should be avoided. If the feature tier could be replaced with other unbiased yet indicative monotonic features, such as historical bar exam passing rates of schools, we believe our performance could be further improved.

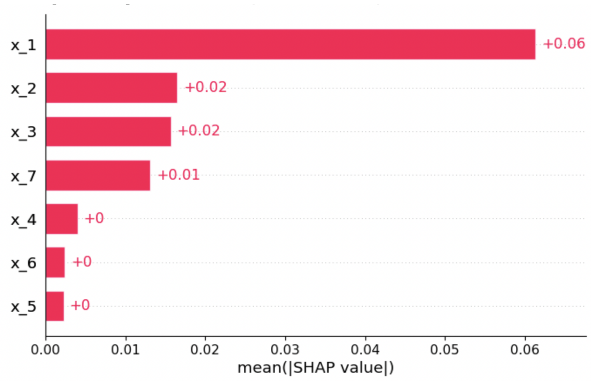


Figure 5: Global feature importance of LawSchool using BShap.

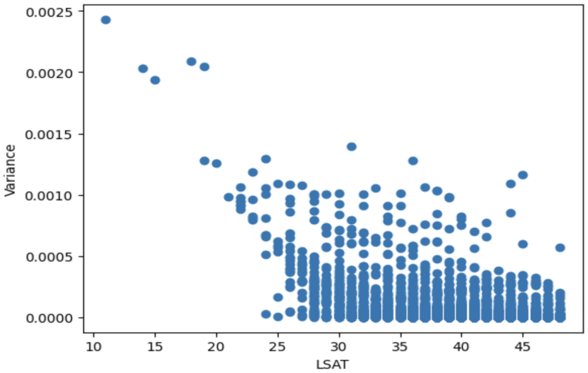


Figure 6: Variance vs LSAT in the Law School dataset.

D.4 Medical - Mammographic Mass

D.4.1 Data description

As a screening tool for breast cancer, mammography is widely used in the medical field. However, due to the uncertainty prediction, biopsies that proved to be benign are not examined as thoroughly as they should be. There has been some previous research relating to semi-supervised learning to reduce the uncertainty of a model by Calderon-Ramirez et al. (2021). On the other hand, our approach reduces the uncertainty of OOD data by finding a lower bound, which is based on the monotonicity property. The data for Mammographic Mass is collected by Elter (2007), for a 5-feature-based classification. Overall, 961 data are available, including 516 benign and 445 malignant. Below are illustrations of all features.

- x_1: BI-RADS assessment, which is a standard assessment used by doctors to describe mammograms. The values range from 1, the benign, to 5, with a high possibility of malignancy.
- x_2: Age
- x_3: Shape of Mammography, classified into four types: round, oval, lobular, and irregular
- x_4: Margin of Mammography, classified into circumscribed, microlobulated, obscured, ill-defined, and spiculated
- x_5: Mammographic density, classified as high, iso, low and fat-containing
- y: The binary label, malignancy=1, benign=0

D.4.2 Model

For MGNAM, we consider the architecture

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_5(x_5). \tag{34}$$

According to the doctor's diagonalization, x_1 is monotonic for predicting the severity of breast cancer. Based on previous research, there is a highly positive relationship between mammographic density and cancer severity, as discussed in Nazari & Mukherjee (2018), therefore, we impose individual monotonicity on both x_1 and x_5.

D.4.3 Results

The AUC of the model is about 90%. The OOD data is more than half due to the rare samples. Therefore, one should be cautious when making predictions based on the model because it is quite uncertain about its predictions. If there is more data available in the hospital, this could be reduced. Based on FMMO, we obtain lower bounds of confidence of approximately 30%, demonstrating the effectiveness of our approach. In the case of healthcare datasets, we believe that it is definitely possible to improve the performance by using more features as well as domain knowledge. Due to the fact that we are not experts in this field, we do not feel comfortable imposing domain knowledge in more complex situations. This is only a simple example provided for demonstration purposes.

E Algorithm Details

E.1 Details for MGNAMs

There are two steps in the training process for the model.

Initially, a mini-batch gradient descent approach with a batch size of 64 is applied to pre-train the model without taking into account monotonicity. The initial learning rate is 1×10^{-2}, and it is multiplied by 0.1 when there is no improvement in the loss over 5 epochs, and early stopping is performed when there is no improvement over 10 epochs.

As part of Step 2, the model is trained to satisfy all monotonicity requirements and the batch gradient descent method is applied. The α factor, which represents the punishment for features exhibiting monotonicity violations, is initially set at 1×10^{-1} and is multiplied by 10 every 10 epochs. At the same time, the learning rate at is set to 1×10^{-2}. Once all monotonies have been satisfied, the remuneration will change to 1×10^{-3} for another 10 epochs of training.

E.2 Details for Baseline Method

We employed the Couenne global optimization solver (Belotti et al. (2009)), which is specifically designed to handle mixed-integer nonlinear optimization (MINLP) problems. This solver effectively minimizes objective functions that involve nonlinear and nonconvex constraints. It uses a reformulation technique that approximates nonconvex problems with linear programming, combined with a branch-and-bound method to manage both continuous and integer variables. Developed through a collaboration between IBM and Carnegie Mellon University, Couenne is available as an open-source package.

For our experiments, we retained the default settings of the Couenne solver across all datasets to ensure consistent results. The experiments were conducted using Pyomo (Hart et al. (2011)), a versatile Python-based open-source tool that supports a wide array of optimization solvers, including Couenne. Pyomo provides a robust framework for formulating, solving, and analyzing optimization problems across different problem classes.

Table 4: OOD results for different ϵ

DATASETS	GMSC	COMPAS	LAW	LIFE-SCIENCE	MAMMOGRAPHY
OOD (%), $\epsilon = 10^{-2}$	8×10^{-3}	0	0	0	6.5
OOD (%), $\epsilon = 10^{-3}$	2.8	10.4	10.5	14.0	52.2
OOD (%), $\epsilon = 10^{-4}$	17.8	95.8	41.8	100	90.0

F Other Discussions

F.1 Binning

By binning or discretizing, continuous features are transformed into discrete ones. Binning is a common practice in numerical partial differential equations (Sethian, 1996). For ML, binning may improve predictive models' accuracy by reducing noise or nonlinearity in the dataset as well as identifying outliers, and invalid and missing values of numerical features. The use of bins is particularly popular in high-risk sectors, where interpretation is of paramount importance. When considering the income feature, for example, people are more likely to consider low-, median-, and high-income classes rather than specific figures. Various binning methods are available, including equal width, equal frequency, and weight of evidence. As the choice of methods depends heavily on the application and the appetite of the user, we will not discuss this further. It is possible to preserve continuous features by leaving original features alone and binning only the features in the optimization process.

F.2 Choice of Models

We apply accurate and transparent monotonic groves of the neural additive model (MGNAM) proposed in Chen & Ye (2023), as three types of monotonicity are included. The code is modified based on the Neural Additive Models (Agarwal et al., 2021). In general, the choice is not unique. Models developed by Liu et al. (2020); Milani Fard et al. (2016); You et al. (2017); Runje & Shankaranarayana (2023) are applicable for individual monotonicity, whereas deep lattice models (Gupta et al., 2020; Cotter et al., 2019) include strong pairwise monotonicity.

F.3 Choice of the Ensemble size M and Threshold ϵ

Through extensive empirical studies on a variety of datasets, Ovadia et al. (2019) suggests an ensemble size of $M = 5$ or 10. To ensure more robust results, we use $M = 10$.

ϵ thresholds are not unique and depend on applications and user preferences. In high-risk sectors, one may choose a very small value for ϵ. However, if the risks are tolerable, a larger ϵ may be chosen. The finance sector, for instance, has seen different types of investors, including risk-averse, risk-neutral, and risk-seeking investors. There is a discussion in Section 3.6 by Petters & Dong (2016).

In this paper, we focus on the choice $\epsilon = 10^{-3}$ such that it works better for all models. We have also tried different values of ϵ. The results for the number of OOD data are recorded in Table 4. When ϵ is very small, given the size of the dataset, the model could be uncertain about a large portion of the dataset, thus making the whole model uncertain. If ϵ is too large, models are already very confident about almost all predictions. The OOD data may not need to be further processed in such extreme cases. As a result, we mainly report the case when $\epsilon = 10^{-3}$. We present some useful cases for other ϵ. The result of GSMC is provided when $\epsilon = 10^{-4}$ since other models are too uncertain. In addition, we provide the results of Mammography when $\epsilon = 10^{-2}$ since other models are very confident. The results are presented in Table 5.

Table 5: Additional results by the FMMO using all monotonicity

DATASETS	ϵ	$\frac{\lvert\mathbb{V}\rvert}{\lvert\mathbb{S}\rvert}$ (%)						**MEAN-ITER**
		$\tau = 0.5$	0.4	0.3	0.2	0.1	0	
GMSC	10^{-4}	100.0	99.8	99.8	99.5	83.3	7.2	63
MAMMOGRAPHY	10^{-2}	94.4	81.5	77.8	72.2	63.0	63.0	9

G Related Work

G.1 Monotonic Models

There has been considerable interest in monotonic models in the ML community. In general, existing methods can be divided into two groups: (1) Monotonicity by constructions (Runje & Shankaranarayana, 2023; You et al., 2017; Milani Fard et al., 2016; Daniels & Velikova, 2010; Sill, 1997) and (2) Monotonicity by regularization (Liu et al., 2020; Bakst et al., 2020; Sivaraman et al., 2020; Sill & Abu-Mostafa, 1996). Both methods have been successful and have been applied to a wide range of applications.

Despite the many successes in monotonic models, these approaches mainly focus on individual monotonicity, and less attention is paid to pairwise monotonicity. A regularization approach is used by Gupta et al. (2020) to enforce individual and strong pairwise monotonicity for linear models, generalized additive models, and the nonlinear function class of multi-layer lattice models. For neural additive models, Chen & Ye (2022) enforces individual and weak pairwise monotonicity through regularization. Using a transparent architecture of MGNAMs, Chen & Ye (2023) enforces individual monotonicity, weak pairwise monotonicity, and strong pairwise monotonicity by regularization. We use MGNAMs here because they incorporate all the monotonicity we require.

G.2 Out-of-Distribution Detection

There have been studies of OOD detection methods that have been successful, including maximum softmax probability (Hendrycks & Gimpel, 2016), temperature scaling (Guo et al., 2017), Monte-Carlo dropout (Gal & Ghahramani, 2016; Srivastava et al., 2015), ensemble methods (Lakshminarayanan et al., 2017), stochastic variational Bayesian inference (Blundell et al., 2015; Graves, 2011; Louizos & Welling, 2017), and approximated Bayesian inference based on the last layer (Riquelme et al., 2018). A detailed comparison of these methods can be found in Ovadia et al. (2019); Yang et al. (2022).

G.3 Fast Marching Methods for the Eikonal Equation

Originally, the FMME (Tsitsiklis, 1995; Sethian, 1996; Helmsen et al., 1996) was proposed to solve the Eikonal equation as follows:

$$
\begin{gathered}
|\nabla u(\mathbf{x})| = \frac{1}{f(\mathbf{x})}, \text{ for } \mathbf{x} \in \Omega, \\
u(\mathbf{x}) = 0, \text{ for } \mathbf{x} \in \partial\Omega,
\end{gathered}
$$

where $|\cdot|$ is the Euclidean norm, ∇ is the gradient, f is given, Ω is the domain, and $\partial\Omega$ is the boundary of the domain.

The following example illustrates how FMME was used in a two-dimensional setting. Assume that the domain has been discretized into a mesh. Meshpoints will be referred to as nodes. Every node (x_i, y_j) has a corresponding value $U_{i,j} = U(x_i, y_j) \approx u(x_i, y_j)$. There are three sets in the algorithm, which are the far set, the considered set, and the accepted set. The far set includes the points that have yet to be calculated, the considered set includes the points that have already been calculated, but do not have the satisfactory solution, and the accepted set contains the points that have the desired solution. Briefly, the algorithm consists of the following steps:

1. In the initialization, assign every node (x_i, y_j) the value of $U_{i,j} = +\infty$ and label them as far; for all nodes $(x_i, y_j) \in \partial\Omega$, set $U_{i,j} = 0$ and label (x_i, y_j) as accepted.

2. For every far node (x_i, y_j), calculate the new value for $\widetilde{U}$ using Eikonal's update formula. If $\widetilde{U} < U_{i,j}$, then set $U_{i,j} = \widetilde{U}$ and label (x_i, y_j) as considered. To be more specific, for the first-order approximation, we have

$$\max\left(D^{-x}_{i,j}U, -D^{+x}_{i,j}U, 0\right)^2 + \max\left(D^{-y}_{i,j}U, -D^{+y}_{i,j}U, 0\right)^2 = \frac{1}{f^2_{i,j}},$$

where

$$D^{\pm x}_{i,j}U = \frac{U_{i\pm1,j} - U_{i,j}}{\pm\Delta x},$$
$$D^{\pm y}_{i,j}U = \frac{U_{i,j\pm1} - U_{i,j}}{\pm\Delta y}.$$

Let

$$U_X = \min\left(U_{i-1,j}, U_{i+1,j}\right),$$
$$U_Y = \min\left(U_{i,j-1}, U_{i,j+1}\right).$$

For sufficient small step size $\Delta x, \Delta y$ such that $\left|\frac{U_X}{\Delta x} - \frac{U_Y}{\Delta y}\right| \leq \frac{1}{f_{i,j}}$, we have the solution to the following quadratic equation

$$\left(\frac{U_{i,j} - U_X}{\Delta x}\right)^2 + \left(\frac{U_{i,j} - U_Y}{\Delta y}\right)^2 = \frac{1}{f^2_{i,j}}.$$

It can be written as $aU^2_{i,j} + bU_{i,j} + c$. The following solution is used due to the physical characteristics of the problem

$$\widetilde{U} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

3. Let $(\widetilde{x}_i, \widetilde{y}_j)$ be the node with the smallest value U in the considered set. Label $(\widetilde{x}_i, \widetilde{y}_j)$ as accepted and remove it from the considered set. **The FMM is utilized here. FMM determines the smallest value by using the heap sort.**

4. For each neighbor (x_i, y_j) of $(\widetilde{x}_i, \widetilde{y}_j)$ that is not accepted, calculate a tentative value $\widetilde{U}$.

5. If $\widetilde{U} < U_{i,j}$, then set $U_{i,j} = \widetilde{U}$. If (x_i, y_j) was previously labeled as far, update the label to be considered.

6. Return to Step 3 if there is a considered node. If not, terminate the process.

In summary, as a result of Eikonal's update formula, the FMME selects the smallest value in each step, resulting in a monotonically nondecreasing sequence as the solution, which is similar to equation 19. The heap algorithm is used to increase the speed of determining the smallest value and removing it from the set.

H Shapley Value

Following Lundstrom et al. (2022), we call the point of interest $\mathbf{x}$ to explain as an explicand and $\mathbf{x}'$ a baseline. The Shapley value (Shapley et al. (1953)) takes as input a set function $v : 2^N \to \mathbb{R}$, which produces attributions s_i for each player $i \in N$ that add up to $v(N)$.

Definition H.1 (Shapley value)**.** *The Shapley value of a player i is given by:*

$$s_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup i) - v(S)).$$

We focus on the Baseline Shapley (BShap) Sundararajan & Najmi (2020), in which

$$v(S) = f(\overline{\mathbf{x}}_S; \mathbf{x}'_{N \setminus S}).$$

That is, baseline values replace the feature's absence. We denote BShap attribution by $\mathrm{BS}_i(\mathbf{x}, \mathbf{x}', f)$ and BS_i sometimes. For example, suppose $f(x_1, x_2) = x_1 + x_2$, $\mathbf{x} = (x_1, x_2)$, $\mathbf{x}' = (0, 0)$, and $S = \{1\}$, then we have $v(S) = f(x_1, 0)$. One common choice of $\mathbf{x}'$ is to take the average of all samples. For the global feature importance of the ith feature, we take the average of the absolute values of BShap

$$\mathcal{A}_i = \frac{1}{n} \sum_{j=1}^{n} |\mathrm{BS}_i(\mathbf{x}_j, \mathbf{x}', f)|.$$