# ACCURACY WITHOUT OPACITY: MONOTONIC TRANSFORMER ADDITIVE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Transformer-based models achieve strong performance on tabular data by learning complex feature interactions, but they remain black boxes and offer no guarantee that learned relationships respect domain-specific monotonicity constraints. Interpretable additive and monotonic models, conversely, provide transparent feature effects but lack the expressiveness to capture rich interactions. To bridge this gap, we introduce the *Monotonic Transformer Additive Model (MTAM)*, which decomposes the model output into additive main effects and feature interactions learned by a Differential Transformer. MTAM addresses monotonicity at three levels: **(a)** the main effects are monotonic by construction; **(b)** the Differential Transformer is adapted to reduce sources of non-monotonicity in the interactions; and **(c)** an augmented Lagrangian scheme imposes monotonicity of the full model as a training constraint and supports an identifiable decomposition through marginal clarity. Across six regression and classification tasks, MTAM narrows the gap between interpretable models and unconstrained transformers while keeping monotonicity violations below $3\%$ on average in public datasets, providing an accurate and transparent alternative to black-box transformers.

## 1 INTRODUCTION

Transformers have demonstrated remarkable success across a broad range of domains, including natural language processing (Vaswani et al., 2017), computer vision (Dosovitskiy et al., 2021), and, more recently, tabular learning (Huang et al., 2020; Gorishniy et al., 2021; Hollmann et al., 2025). Despite their strong predictive performance, interpreting their decisions remains challenging. In tabular settings, attention layers repeatedly mix information across features, making the contribution of any individual feature difficult to disentangle. Moreover, attention weights have been shown to be unreliable indicators of model reasoning (Jain & Wallace, 2019), limiting their usefulness as explanations. This lack of transparency hinders the adoption of transformer-based models in high-stakes domains, where decision-making processes must be interpretable and auditable. In many such applications, prior knowledge further dictates that predictions should vary monotonically with respect to specific features, and violations of these known relationships can undermine trust regardless of predictive accuracy (Chen & Ye, 2023). While post-hoc explanation methods such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) can provide feature attributions for individual predictions, they operate independently of the model's learning process and may not faithfully reflect the computations driving its outputs (Slack et al., 2020; Rudin, 2019). Furthermore, they offer no guarantees regarding the enforcement of monotonic constraints.

In this paper, we introduce the **Monotonic Transformer Additive Model (MTAM)**, a deep learning architecture that addresses a key limitation of existing models: the inability to simultaneously provide expressive interaction modeling, monotonicity guarantees, identifiable additive decomposition, and competitive predictive performance. While additive models can offer transparent feature-wise explanations and support monotonic constraints, they are often limited in their ability to capture complex interactions. Conversely, transformer-based models can learn rich interaction structures, but their representations are generally neither interpretable nor compatible with monotonicity requirements. MTAM bridges this gap by embedding transformer-based interaction learning within a monotonic additive framework. Main effects are modeled using constrained monotonic networks, which guarantee monotonicity by construction, while interactions are learned through a monotonicity-oriented Differential Transformer and enhanced through an augmented Lagrangian



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

objective. Rather than treating monotonicity as a purely optimization-level constraint, MTAM systematically removes architectural sources of non-monotonic behavior through non-negative feature embeddings, the removal of layer normalization, and monotonic feed-forward networks. These design choices act as inductive biases that allow for monotonic interaction learning without restricting the transformer's capabilities. The resulting architecture closes the gap between interpretable monotonic models and expressive transformer-based interaction learners, enabling rich feature interaction modeling while preserving both interpretability and domain-consistent behavior.

Our main contributions are as follows:

- **Monotonic Transformer Additive Model (MTAM).** We introduce MTAM, a transformer-based additive model for tabular learning that combines the interpretability of additive decompositions with the expressive modeling capacity of attention-based feature interactions. MTAM decomposes predictions into main and interaction effects while enabling monotonicity constraints to be incorporated throughout the model.
- **Monotonic Differential Transformer for expressive feature interactions.** We develop a monotonicity-oriented Differential Transformer that learns selective and interpretable feature interactions while remaining compatible with monotonicity requirements.
- **Marginal clarity for identifiable interaction learning.** We introduce a marginal clarity regularization mechanism that promotes a principled separation between main and interaction effects, improving the interpretability of the learned decomposition.
- **Strong empirical end-to-end monotonicity enforcement.** We develop an end-to-end framework for monotonic learning that combines constrained monotonic networks, monotonic interaction modeling, and augmented Lagrangian optimization, achieving competitive performance with unconstrained models, while preserving monotonicity.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 and 4 introduce MTAM. Section 5 presents the experimental results, and Section 6 concludes the paper.

## 2 Related work

MTAM builds on three complementary research directions: interpretable additive neural models, attention-based modeling of feature interactions, and monotonic neural networks.

**Decomposable Neural Models.** Neural Additive Models (NAMs) (Agarwal et al., 2021) learn a separate shape function for each feature, yielding an exact decomposition of predictions into additive feature-wise effects. GAMI-Net (Yang et al., 2021) extends this framework with sparse pairwise interaction subnetworks and introduces marginal clarity constraints to separate main effects from interactions. These ideas are closely related to the identifiability properties of functional ANOVA decompositions (Lengerich et al., 2019). More recently, Thielmann et al. (2026) proposed a decomposition for tabular transformers that separates additive marginal effects from a transformer-based interaction component, combining interpretability with efficient interaction modeling. MTAM adopts this decomposition and augments it with Differential Transformer-based interactions (Ye et al., 2025).

**Interpretable feature interactions.** Transformers achieve strong performance on tabular data (Huang et al., 2020; Gorishniy et al., 2021; Hollmann et al., 2025) largely due to their ability to model feature dependencies. However, conventional attention often produces dense interaction patterns that are difficult to interpret. Additive models improve interpretability by restricting interactions to a small set of explicit terms (Yang et al., 2021), but are typically limited to low-order dependencies. Differential Transformers (Ye et al., 2025) offer an alternative by subtracting two attention maps, suppressing common-mode attention and producing sparser, more selective interaction patterns. MTAM replaces the interaction module of Thielmann et al. (2026) with a monotonicity-preserving Differential Attention mechanism while maintaining a clear separation between marginal and interaction effects.

**Monotonic Neural Networks.** A separate line of work incorporates prior knowledge about the direction of feature effects through monotonicity constraints, using sign-constrained networks (Sill, 1998), lattice models (You et al., 2017), post-hoc certification (Liu et al., 2020), counterexample-guided training (Sivaraman et al., 2020), and Lipschitz- or weight-constrained architectures (Nolte

et al., 2023; Runje & Shankaranarayana, 2023; Kim & Lee, 2024). Sartor et al. (2025) show that suitably constrained multilayer perceptrons are universal approximators of monotone functions, providing both theoretical guarantees and a practical way to enforce monotonicity during training.

MTAM integrates monotonicity at three levels. First, the main effects use constrained monotonic neural networks, upholding monotonicity by construction. Second, the interaction component uses a monotonicity-oriented modification of the Differential Transformer, based on non-negative embeddings, the removal of layer normalization, and monotonic feed-forward blocks. Third, an augmented Lagrangian imposes monotonicity of the full model as a training constraint, together with marginal clarity. To the best of our knowledge, no existing method combines monotonicity constraints with attention-based interactions and an identifiable additive decomposition.

# 3 Monotonic Transformer Additive Model

## 3.1 Problem setup

We use capital letters for random variables and lowercase letters for their realizations. Let $\mathbf{X} = (X_1, \dots, X_K)$ be a random vector of features with values in $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K \subseteq \mathbb{R}^K$, let $\mathbf{x} = (x_1, \dots, x_K)$ denote a realization, and let $Y$ be the target. We consider two disjoint index sets $\mathcal{M}^+, \mathcal{M}^- \subseteq \{1, \dots, K\}$ of features for which the prediction is required to be non-decreasing and non-increasing, respectively. We write $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$ for the constrained features and $\mathcal{U} = \{1, \dots, K\} \setminus \mathcal{M}$ for the unconstrained features. We use $\mathbf{x}_\mathcal{U}$ and $\mathbf{x}_{-j}$ to denote the sub-vectors indexed by $\mathcal{U}$ and by all indices except $j$, respectively.

**Definition 1.** A function $l : \mathcal{X} \to \mathbb{R}$ is *monotonic with respect to* $(\mathcal{M}^+, \mathcal{M}^-)$ if, for every $j \in \mathcal{M}$, every $\mathbf{x}_{-j}$, and all $x_j \le x'_j$,

$$l(x_j, \mathbf{x}_{-j}) \le l(x'_j, \mathbf{x}_{-j}) \quad \text{if } j \in \mathcal{M}^+, \quad \text{or} \quad l(x_j, \mathbf{x}_{-j}) \ge l(x'_j, \mathbf{x}_{-j}) \quad \text{if } j \in \mathcal{M}^-.$$

This is weak monotonicity, allowing $l$ to remain constant but not to move in the undesired direction. Without loss of generality, we transform every feature $j \in \mathcal{M}^-$ as $x_j \mapsto -x_j$, so that all constrained features are required to be non-decreasing. We therefore assume $\mathcal{M}^- = \emptyset$ below.

**Model.** MTAM considers the following additive structure:

$$l(\mathbf{x}) = g(\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]) = \beta + \sum_{j=1}^{K} f_j(x_j) + h(\mathbf{x}), \qquad \mathbb{E}[f_j(X_j)] = 0, \quad \mathbb{E}[h(\mathbf{X})] = 0, \tag{1}$$

where $g$ is a strictly increasing link function, $\beta$ is an intercept, $f_j$ are univariate main effects, and $h$ is an interaction component. Equivalently, $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}] = g^{-1}\big(l(\mathbf{x})\big)$. Since $g^{-1}$ is strictly increasing, monotonicity of $l$ is equivalent to monotonicity of the conditional mean.

We seek a model of the form equation 1 that satisfies three requirements: **(R1)** *monotonicity*: $l$ is monotonic in the sense of Definition 1; **(R2)** *marginal clarity*: additive and interaction effects are identifiable, with $h$ containing no contribution attributable solely to a single feature; **(R3)** *expressiveness*: $h$ captures only higher-order dependencies between features, emphasizing interaction-relevant signals while suppressing common-mode patterns.

**Marginal Clarity.** For **(R2)**, we adopt the functional ANOVA perspective (Lengerich et al., 2019). We say that $h$ satisfies *marginal clarity* (Yang et al., 2021) if

$$\mathbb{E}[h(\mathbf{X}) \mid X_k] = 0 \quad \text{almost surely for every } k = 1, \dots, K. \tag{2}$$

Thus, $h$ contains no component attributable solely to any individual feature. In particular, marginal clarity implies

$$\mathbb{E}\big[f_k(X_k)\, h(\mathbf{X})\big] = \mathbb{E}[f_k(X_k)\, \mathbb{E}[h(\mathbf{X}) \mid X_k]] = 0, \qquad k = 1, \dots, K, \tag{3}$$

so the interaction component is orthogonal to every main effect.

# 4 MTAM Components

MTAM consists of two components *main-effect networks*, which model the effect of each feature on its own, and a *Differential Transformer*, which models the interactions between features. The

main effects of constrained features are monotonic by construction. The monotonicity of the model output is addressed during training through an augmented Lagrangian.

**Main effect Networks.** Following Neural Additive Models (Agarwal et al., 2021), each main effect $f_j$ is a small neural network that receives only its own feature $x_j$ and outputs its contribution $f_j(x_j)$ to the prediction. Since $f_j$ does not depend on any other feature, it can be visualized directly. For unconstrained features $j \in \mathcal{U}$, $f_j$ is a standard MLP. For constrained features $j \in \mathcal{M}$, $f_j$ is a constrained monotonic network (Sartor et al., 2025): every layer computes

$$\mathbf{z}^{(l+1)} = \sigma_l\big(\mathbf{W}_+^{(l)}\,\mathbf{z}^{(l)} + \mathbf{b}^{(l)}\big), \qquad \mathbf{W}_+^{(l)} = \big|\mathbf{V}^{(l)}\big|, \tag{4}$$

where the weights $\mathbf{W}_+^{(l)}$ are element-wise non-negative, $\mathbf{V}^{(l)}$ are the unconstrained trainable parameters, and $\sigma_l$ are non-decreasing activation functions. Each $f_j$ is centered by subtracting its empirical mean over the training data, a constant shift that preserves monotonicity and ensures a zero empirical mean. In our work, we assume that the main effects are centered, $\mathbb{E}[f_j(X_j)] = 0$.

**Interaction Component.** Following common practice in tabular transformers (Gorishniy et al., 2021; Thielmann et al., 2026), each feature $k$ is represented by a token

$$\mathbf{t}_k^{(0)} = \mathbf{c}_k + E_k(x_k), \tag{5}$$

where $\mathbf{c}_k$ is a learned feature-specific embedding and $E_k$ is a feature-value embedding. The tokens are processed by $L$ layers of a Differential Transformer (Ye et al., 2025), and the resulting token states are aggregated into the interaction component $h(\mathbf{x})$. Each layer uses Differential Attention,

$$\mathrm{DiffAttn}(\mathbf{T}) = \Big[\,\mathrm{softmax}\Big(\frac{\mathbf{Q}_1\mathbf{K}_1^\top}{\sqrt{d}}\Big) - \lambda\,\mathrm{softmax}\Big(\frac{\mathbf{Q}_2\mathbf{K}_2^\top}{\sqrt{d}}\Big)\Big]\mathbf{V}, \tag{6}$$

where $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{K}_1, \mathbf{K}_2, \mathbf{V}$ are linear projections of the token matrix $\mathbf{T}$, $d$ is the head dimension, and $\lambda$ is a learnable scalar. Subtracting the two attention is designed to cancel common-mode attention noise and encourages sparse attention patterns (Ye et al., 2025).

**Monotonicity of the interaction component.** The attention-based interaction component $h$ is in general not monotonic in the constrained features. Softmax attention weights sum to one, so increasing one feature shifts attention away from the others, and Differential Attention can additionally produce negative weights. Either effect can make $h$ decrease when a constrained feature increases.

Requiring $h$ itself to be monotonic would, however, conflict with marginal clarity. Suppose $h$ is non-decreasing in a constrained feature $x_j$. Marginal clarity requires $\mathbb{E}[h(\mathbf{X}) \mid X_j] = 0$, i.e., $h$ averages to zero at every value of $x_j$. A function that never decreases in $x_j$ yet averages to zero at every value of $x_j$ cannot change with $x_j$ at all. Imposing both constraints would therefore force $h$ to ignore $x_j$, and constrained features could not take part in any interaction.

MTAM therefore does not require $h$ to be monotonic. Instead, it imposes monotonicity on the *total effect* $f_j(x_j) + h(\mathbf{x}), \forall j \in \mathcal{M}$. To make this constraint easier to satisfy, we modify the Differential Attention mechanism for constrained features so that the interaction component cannot overturn their monotonic main effects, while $h$ may still be non-monotonic on its own.

### 4.1 Training with an augmented Lagrangian

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the training set of $n$ samples, where $\mathbf{x}_i$ is the feature vector of sample $i$ and $x_{ik}$ is the value of its $k$-th feature. Two families of constraints are imposed during training.

**Marginal Clarity.** We impose orthogonality of the interaction component to every main effect. Since the expectation in equation 3 is unknown, we replace it by its empirical average over the training set:

$$o_k(\theta) = \frac{1}{n}\Big|\sum_{i=1}^{n} f_k(x_{ik})\,h(\mathbf{x}_i)\Big| = 0, \qquad k = 1, \ldots, K. \tag{7}$$

**Monotonicity.** For monotonicity, as indicated in Section 3.1, all constrained features are required to be non-decreasing. Since $l_\theta$ is differentiable, monotonicity in $x_j$ is equivalent to the total effect

$f_j(x_j) + h(\mathbf{x})$ having a non-negative derivative, i.e., $\partial_{x_j} l_\theta(\mathbf{x}) \geq 0$ for every $j \in \mathcal{M}$. We measure its violation by

$$v_j(\theta) = \mathbb{E}_{\mathbf{x}\sim Q}\Big[\big[-\partial_{x_j} l_\theta(\mathbf{x})\big]_+\Big], \qquad j \in \mathcal{M}, \tag{8}$$

where $[\cdot]_+ = \max(\cdot, 0)$ and $Q$ is a distribution over inputs that mixes the training data with points sampled uniformly from the bounding box $\mathcal{S}$ of the data, so that monotonicity is also enforced in regions not covered by the training data. Since $v_j \geq 0$, the inequality constraint reduces to the equality constraint $v_j(\theta) = 0$.

**Optimization.** Given the training set $\mathcal{D}$, a loss $\ell$ (e.g., squared error or cross-entropy), and parameters $\theta$, we solve

$$\begin{aligned} &\min_\theta \mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, l_\theta(\mathbf{x}_i)\big) + R(\theta) \\ &\text{s.t. } o_k(\theta) = 0 \quad (k = 1, \ldots, K), \qquad v_j(\theta) = 0 \quad (j \in \mathcal{M}), \end{aligned} \tag{9}$$

where $R$ collects standard regularizers. We solve equation 9 with the augmented Lagrangian method,

$$\mathcal{L}_\mu(\theta, \boldsymbol{\alpha}, \boldsymbol{\nu}) = \mathcal{L}(\theta) + \sum_{k=1}^{K}\Big(\alpha_k o_k(\theta) + \frac{\mu}{2} o_k(\theta)^2\Big) + \sum_{j\in\mathcal{M}}\Big(\nu_j v_j(\theta) + \frac{\mu}{2} v_j(\theta)^2\Big), \tag{10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ and $\boldsymbol{\nu} = (\nu_j)_{j\in\mathcal{M}}$ are the Lagrange multipliers of the marginal clarity and monotonicity constraints, respectively, and $\mu > 0$ is the penalty parameter. Training alternates stochastic gradient steps on $\theta$ with the multiplier updates $\alpha_k \leftarrow \alpha_k + \mu\, o_k(\theta)$ and $\nu_j \leftarrow \nu_j + \mu\, v_j(\theta)$.

#### 4.1.1 Monotonic Inductive Biases for Transformers

Guaranteeing monotonicity architecturally is difficult for transformers. However, what matters is that the interaction component $h$ does not overturn the monotonic main effects (Section 3.1). We find that several design choices substantially reduce monotonicity violations of the full model without noticeably affecting predictive performance (Section 5).

**Monotonic uncontextualized embeddings.** A key contribution of Thielmann et al. (2026) is the use of uncontextualized embeddings as direct representations of individual features. These embeddings remain central to MTAM, but constrained features require an additional inductive bias: their embeddings must preserve the ordering of the inputs. We therefore parameterize the embedding of each constrained feature with non-negative weights, so that every embedding coordinate is non-decreasing in $x_j$. As a result, increasing a constrained feature cannot reverse the direction of its latent representation before it is processed by the monotonic NAM components or the transformer layers, promoting monotonic behavior from the earliest stage of the model.

**Removing layer normalization.** Layer normalization rescales representations using statistics computed jointly across embedding dimensions. Since these statistics depend on the input features, they can flip the sign of the derivatives with respect to constrained features. Removing layer normalization reduced monotonicity violations in our experiments (Section 5).

**Monotonic feed-forward blocks.** Finally, we replace the feed-forward blocks within each transformer layer by monotonic MLPs. This does not guarantee monotonicity of the transformer layer as a whole, since attention can still reverse the direction of a feature's effect, but it adds an inductive bias that empirically improves constraint satisfaction.

In MTAM, monotonicity is thus handled at three levels: the NAM components are monotonic by construction, the inductive biases above keep the transformer from overturning them, and the augmented Lagrangian enforces monotonicity of the full model during training.

## 5 Experiments

We evaluate MTAM through a comprehensive set of synthetic and real-world experiments designed to assess predictive performance, interaction interpretability, identifiability, and monotonicity satisfaction. Our evaluation consists of two complementary parts. In Section 5.1, we conduct a controlled

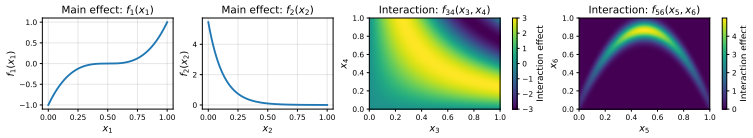


Figure 1: Ground Truth of the synthetic function in equation 11.

ablation study to evaluate the contribution of each proposed component to the model's predictive performance, interpretability, and monotonicity satisfaction. In Section 5.2 , we further evaluate MTAM on a collection of public regression and classification benchmarks, including AutoMPG[1], California Housing[2], Diamonds[3], Heart Disease[4], and Loan Defaulter[5], together with a Cloud Resource Management dataset (see Appendix A.1). Across these datasets, we compare both constrained and unconstrained variants of our method against representative interpretable and black-box baselines, including NBM (Radenovic et al., 2022), MLP (Sartor et al., 2025), FT-Transformer (Gorishniy et al., 2021), and NAMformer (Thielmann et al., 2026).

All models are trained using the same architecture and optimization settings across every dataset (see Appendix A.2). We do not perform dataset-specific hyperparameter tuning to provide a fair comparison between methods. All reported results are obtained from 10 independent random seeds, and we report the mean and standard deviation for every metric. For regression tasks, predictive performance is measured using the coefficient of determination ($R^2$), while classification performance is evaluated using the area under the ROC curve (AUC). For monotonic models, we additionally report monotonicity satisfaction (%). This metric is estimated by sampling instances from the test set and, for each constrained feature, varying only that feature while keeping all remaining variables fixed. Specifically, we generate 1k evaluation points for continuous features (or all possible values for discrete features) within the feature domain and measure the predictions that satisfy the monotonic relationship. Unless explicitly indicated otherwise, all monotonicity constraints are defined based on prior domain knowledge and are applied consistently across all methods.

### 5.1 Ablation Study

We begin by evaluating the individual components of MTAM on a controlled synthetic benchmark. Following Yang et al. (2021), we construct a six-dimensional regression problem and modify it to include both monotonic and interaction effects. In particular, $x_1$ contributes through a non-decreasing cubic main effect, while $x_2$ contributes through a non-increasing exponential main effect. The remaining variables participate exclusively in higher-order interactions, yielding the target function

$$y = 8(x_1 - \tfrac{1}{2})^3 + \tfrac{1}{10}e^{-8x_2+4} + 3\sin(2\pi x_3 x_4) + 5e^{-2(2x_5-1)^2 - \frac{1}{2}\left[15x_6 + 12(2x_5-1)^2 - 13\right]^2} + \epsilon. \tag{11}$$

We generate 10k samples by independently drawing each feature from a uniform distribution on $[0, 1]$. To further assess robustness and feature-selection behavior, we augment the dataset with 24 additional nuisance variables, $(x_7, \ldots, x_{30})$, that are sampled independently and have no influence on the target variable. Consequently, an ideal model should allocate its additive capacity primarily to $x_1$ and $x_2$, recover the interaction structure between $(x_3, x_4)$ and $(x_5, x_6)$, and ignore the remaining noisy features. Figure 1 visualizes the ground-truth main effects and interaction surfaces.

#### 5.1.1 Impact of Differential Transformer

To better understand the role of Differential Attention, we train two MTAM variants on the synthetic dataset: one using standard softmax attention and one using Differential Attention. We then analyze the final-layer attention weights emitted by the [CLS] token (Figure 2). Since the [CLS] token aggregates information used by the transformer interaction module to form the prediction representation, its attention pattern reveals which features contribute most strongly to modeling interactions (Gorishniy et al., 2021). In this setting, the interaction branch should primarily focus on

[1] https://archive.ics.uci.edu/ml/datasets/auto+mpg

[2] https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html

[3] https://www.openml.org/search?type=data&sort=runs&id=42225&status=active

[4] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[5] https://github.com/retna319/SMNN/blob/main/data/realworld-datasets/loan_preprocessed.csv

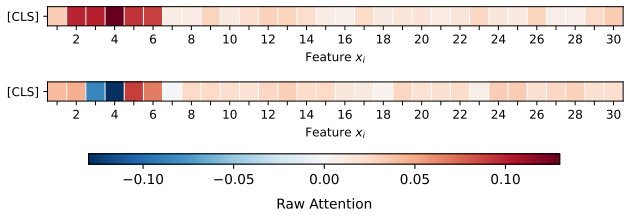


Figure 2: Average attention maps for softmax attention (top) and Differential Attention (bottom).

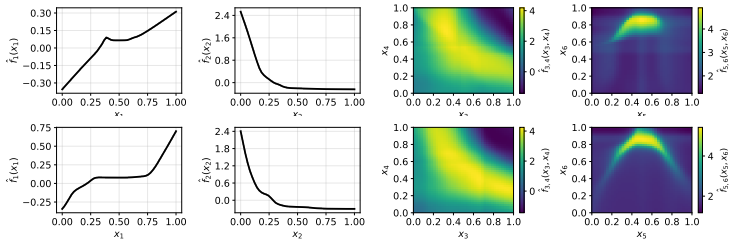

Figure 3: Learned Main and Interaction Effects for NAMformer (top) and MTAM (bottom).

features 3 through 6, which are the only variables participating in higher-order dependencies, while features 1 and 2 should be handled largely by the additive component. Standard attention only partially recovers this structure (Figure 2, top). Although it assigns the largest weights to the predictive features, it does not clearly distinguish variables involved in interactions from those contributing only through marginal effects. Features 2 through 6 receive similar attention values ($\approx 0.10$), making it difficult to infer which variables are actually responsible for the interaction behavior captured by the transformer.

Differential Attention produces a substantially clearer pattern (Figure 2, bottom). While predictive features still receive the largest magnitudes, attention becomes concentrated on the interaction variables: features 3 and 4 receive large negative weights ($\approx -0.10$), whereas features 5 and 6 receive large positive weights ($\approx 0.08$). In contrast, the marginal effect variables $x_1$ and $x_2$ receive considerably smaller weights ($\approx 0.04$). This behavior suggests that Differential Attention endows the interaction module with a more expressive mechanism for representing feature dependencies. Rather than merely identifying predictive variables, it can emphasize and contrast distinct groups of features according to their functional role, separating interaction-driven effects from additive ones while simultaneously grouping related interaction partners. As a result, the learned attention maps more faithfully reflect the underlying interaction structure of the data and provide a more interpretable view of how interactions are represented.

### 5.1.2 Identifiability

To evaluate the impact of the proposed identifiability mechanism, we compare two variants of MTAM: (i) without the orthogonality, and (ii) with the proposed orthogonality constraint. Both models employ the dropout-based identifiability mechanism proposed in Thielmann et al. (2026) with $p_{\text{dropout}} = 0.3$, while all monotonicity constraints are disabled. The goal is to evaluate whether enforcing orthogonality between the additive and interaction branches yields a cleaner recovery of the true main and interaction effects.

Figure 3 compares the learned decompositions of the unconstrained (top) and constrained (bottom) MTAM models. Without orthogonality, the model correctly identifies the interaction pairs $(x_3, x_4)$ and $(x_5, x_6)$, but the recovered effects are diffuse. While $\hat{f}_{3,4}$ remains reasonably interpretable, $\hat{f}_{5,6}$ fails to clearly capture the interaction structure. A similar pattern is observed in the main effects: $\hat{f}_2$ is recovered accurately, whereas $\hat{f}_1$ only partially captures the underlying trend and exhibits noticeable distortions. In contrast, enforcing orthogonality produces substantially cleaner decompositions. The constrained model accurately recovers the cubic shape of $f_1$ and the decreasing exponential behavior of $f_2$, while both interaction surfaces become sharper and more closely resemble the ground truth. In particular, $\hat{f}_{3,4}$ is less diffuse and $\hat{f}_{5,6}$ becomes clearly identifiable, demonstrating a more faithful separation of marginal and interaction effects.

Table 1: Constraint satisfaction (%) across regression datasets. Reported as (mean ± std. dev.).

| | California | Diamonds | | | | | |
|---|---|---|---|---|---|---|---|
| | MedInc | Carat | Color | Clarity | X | Y | Z |
| $\text{MTAM}_{\text{mono}}$ | 99.227 ±2.237 | 100.000 ±0.000 | 100.000 ±0.000 | 100.000 ±0.000 | 100.000 ±0.000 | 100.000 ±0.000 | 100.000 ±0.000 |
| MTAM | 94.544 ±8.239 | 82.402 ±20.833 | 90.758 ±8.096 | 86.558 ±9.947 | 63.854 ±10.998 | 59.852 ±31.709 | 54.277 ±26.006 |

Table 2: Constraint satisfaction (%) across regression datasets. Reported as (mean ± std. dev.).

| | Autompg | | | Cloud Data | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Horsepower | Weight | Displacement | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $R_0$ | $R_1$ |
| $\text{MTAM}_{\text{mono}}$ | 94.75 ±4.076 | 93.406 ±4.338 | 95.12 ±4.63 | 99.965 ±0.111 | 80.554 ±12.140 | 98.829 ±1.484 | 60.691 ±16.396 | 54.254 ±11.450 | 25.088 ±16.237 |
| MTAM | 90.454 ±11.364 | 92.207 ±8.215 | 81.753 ±11.294 | 96.946 ±5.241 | 49.302 ±17.532 | 95.071 ±3.654 | 35.261 ±8.314 | 27.198 ±10.097 | 3.381 ±2.794 |

Table 3: Constraint satisfaction (%) across classification datasets. Reported as (mean ± std. dev.).

| | Heart | | Loan | | | | |
|---|---|---|---|---|---|---|---|
| | Trestbps | Chol | Credit Score | Bankruptcies | Employment Length | Annual Income | DtI Ratio |
| $\text{MTAM}_{\text{mono}}$ | 92.613 ±7.080 | 94.180 ±6.031 | 99.978 ±0.045 | 98.908 ±0.995 | 86.945 ±11.490 | 99.925 ±0.114 | 99.978 ±0.039 |
| MTAM | 76.947 ±14.158 | 64.317 ±8.825 | 99.983 ±0.024 | 85.105 ±14.578 | 72.023 ±8.718 | 70.97 ±40.219 | 51.825 ±44.366 |

These observations align with the theoretical motivation of upholding marginal clarity (equation 7). By enforcing the purity principle underlying functional ANOVA decompositions, the orthogonality constraint encourages the additive branch to explain low-order structure and prevents interaction capacity from being spent on marginal effects. As a result, the learned decomposition becomes more identifiable and the interaction module can focus on modeling genuine feature dependencies. This separation also promotes monotonic learning: because monotonic trends are primarily and more easily captured by the additive neural branch, the interaction component is less likely to counteract them, making global monotonicity easier to satisfy while preserving expressive interaction modeling.

### 5.1.3 Monotonic Inductive Biases for the Differential Transformer

Monotonicity in MTAM is enforced through three complementary mechanisms: (i) monotonic NAM components, (ii) global monotonicity constraints optimized via an augmented Lagrangian, and (iii) monotonic inductive biases in the transformer interaction module. In this section, we isolate and evaluate the contribution of the transformer-specific inductive biases to constraint satisfaction. To this end, we compare the full MTAM model presented in Section 3 against a variant without the inductive biases introduced in Section 4.1.1. Moving beyond the toy example, we conduct experiments on the real-world regression and classification datasets described previously.

Tables 1-3 show a consistent improvement in monotonicity satisfaction when the proposed transformer inductive biases are enabled. Across most datasets, the full MTAM achieves either perfect or near-perfect constraint satisfaction, while the unconstrained variant exhibits substantially lower satisfaction rates and considerably higher variance across runs. The effect is particularly pronounced on datasets with multiple constrained variables and strong feature interactions. The classification results exhibit the same overall pattern. On both the Heart and Loan datasets, introducing inductive biases substantially improves constraint satisfaction for monotonic features while also reducing variability across runs, indicating more stable optimization and more consistent adherence to the monotonic relationships.

The Cloud Resource Management dataset further highlights the importance of the architectural constraints in realistic large-scale settings. Unlike the public benchmarks, this dataset contains multiple monotonic resource allocation and throughput variables that interact strongly with one another, creating a significantly more challenging environment to maintain monotonic behavior. Despite this increased complexity, MTAM consistently achieves higher monotonicity satisfaction across all six constrained variables, even in those cases where exact monotonic behavior is difficult to achieve.

## 5.2 Benchmark Evaluation

Having validated the proposed architecture in the previous subsection, we now compare MTAM against both interpretable and black-box baselines on real-world datasets. Table 4 reveals a clear trade-off between predictive performance, interpretability, and monotonicity enforcement. Unconstrained black-box models, particularly FT-Transformer, often achieve the highest predictive accuracy, benefiting from unrestricted interaction modeling. However, these gains come at a cost: no interpretability nor consistency with known monotonic relationships.

Table 4: Model performance comparison across datasets.

| | | Regression ↑ ($R^2$) | | | | Classification ↑ (AUC) | |
|---|---|---|---|---|---|---|---|
| Category | Model | AutoMPG | California | Diamond | Cloud | Heart | Loan |
| Interpretable | $\text{NBM}_{\text{mono}}$ | $0.813 \pm 0.044$ | $0.624 \pm 0.013$ | $0.895 \pm 0.017$ | $0.688 \pm 0.037$ | $0.897 \pm 0.045$ | $0.707 \pm 0.002$ |
| | NBM | $0.759 \pm 0.062$ | $0.631 \pm 0.009$ | $0.895 \pm 0.017$ | $0.768 \pm 0.044$ | $0.892 \pm 0.054$ | $0.707 \pm 0.002$ |
| | NAMformer | $0.810 \pm 0.065$ | $0.781 \pm 0.011$ | $0.959 \pm 0.011$ | $0.811 \pm 0.052$ | $0.906 \pm 0.046$ | $0.708 \pm 0.002$ |
| Black-box | $\text{MLP}_{\text{mono}}$ | $0.833 \pm 0.056$ | $0.798 \pm 0.013$ | $0.930 \pm 0.062$ | $0.749 \pm 0.046$ | $0.874 \pm 0.065$ | $0.708 \pm 0.002$ |
| | MLP | $0.809 \pm 0.065$ | $0.796 \pm 0.011$ | $0.929 \pm 0.067$ | $0.769 \pm 0.067$ | $0.907 \pm 0.055$ | $0.709 \pm 0.002$ |
| | FT-Transformer | $0.871 \pm 0.033$ | $0.819 \pm 0.015$ | $0.981 \pm 0.001$ | $0.857 \pm 0.049$ | $0.880 \pm 0.047$ | $0.709 \pm 0.002$ |
| Ours | $\text{MTAM}_{\text{mono}}$ | $0.841 \pm 0.024$ | $0.772 \pm 0.012$ | $0.888 \pm 0.044$ | $0.753 \pm 0.059$ | $0.887 \pm 0.025$ | $0.708 \pm 0.001$ |
| | MTAM | $0.860 \pm 0.022$ | $0.800 \pm 0.009$ | $0.951 \pm 0.019$ | $0.815 \pm 0.052$ | $0.910 \pm 0.024$ | $0.709 \pm 0.001$ |

Table 5: Average monotonicity satisfaction (%) for monotonic features across datasets.

| | | Regression | | | | Classification | |
|---|---|---|---|---|---|---|---|
| Category | Model | AutoMPG | California | Diamond | Cloud | Heart | Loan |
| Interpretable | $\text{NBM}_{\text{mono}}$ | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| | NBM | 91.63 | 94.65 | 99.43 | 64.73 | 61.21 | 96.10 |
| | NAMformer | 68.56 | 94.24 | 94.81 | 51.36 | 76.29 | 85.39 |
| Black-box | $\text{MLP}_{\text{mono}}$ | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| | MLP | 65.24 | 87.62 | 87.84 | 52.89 | 61.98 | 69.59 |
| | FT-Transformer | 89.07 | 84.82 | 70.08 | 47.32 | 66.79 | 71.81 |
| Ours | $\text{MTAM}_{\text{mono}}$ | 94.43 | 99.23 | 100.00 | 69.89 | 92.61 | 97.15 |
| | MTAM | 88.14 | 94.54 | 72.95 | 51.93 | 70.63 | 75.98 |

Among interpretable methods, explicitly modeling feature interactions is crucial. NAMformer consistently outperforms additive NBM variants, demonstrating that additive explanations alone are insufficient for capturing complex real-world dependencies. Nevertheless, despite its improved predictive performance, NAMformer does not provide monotonic guarantees and exhibits substantial monotonicity violations on several datasets (Table 5).

MTAM is designed to balance these competing objectives. Compared to interpretable baselines, it consistently improves predictive performance, outperforming NBM and matching or exceeding NAMformer on most datasets. More importantly, MTAM remains highly competitive with unrestricted black-box models despite enforcing an identifiable additive decomposition and monotonic structure. While FT-Transformer achieves the best results, the performance gap is generally modest, indicating that these structural constraints can be incorporated with limited loss in accuracy.

The effect of monotonicity constraints is further illustrated by combining the predictive and monotonicity results in Tables 4 and 5. Across architectures, enforcing monotonicity introduces only a moderate reduction in predictive performance. For MTAM in particular, $\text{MTAM}_{\text{mono}}$ remains competitive while achieving strong monotonicity compliance, with satisfaction rates exceeding $90\%$ on most datasets and approaching perfect compliance on several of them. While architecturally constrained models such as $\text{NBM}_{\text{mono}}$ and $\text{MLP}_{\text{mono}}$ achieve perfect satisfaction rates by construction, they either exhibit weaker predictive performance or do not provide interpretable interaction modeling. In contrast, unconstrained models often achieve strong predictive accuracy but display substantial monotonicity violations. $\text{MTAM}_{\text{mono}}$ occupies a favorable middle ground: it preserves an interpretable additive decomposition with explicit interaction modeling, achieves near-perfect monotonicity satisfaction, and remains competitive with the strongest unrestricted models. Consequently, among models that simultaneously satisfy monotonicity constraints and provide interpretable feature and interaction effects, $\text{MTAM}_{\text{mono}}$ delivers the strongest overall predictive performance.

## 6 Conclusions

We introduced Monotonic Transformer Additive Models (MTAM), a monotonic additive architecture that combines intelligible feature effects with interpretable transformer-based interaction modeling. The central finding of this work is that expressive attention-based interactions can coexist with monotonic constraints when the model is structured to separate and identify main and interaction effects. Empirically, MTAM achieves a favorable balance between predictive performance, monotonicity satisfaction, and interpretability across a range of datasets, suggesting a promising direction for bringing modern transformer architectures into high-stakes settings where transparency and domain-compliant behavior are essential.

### AI USE STATEMENT

In this work, we used generative AI tools solely to assist with writing and language editing, including improving clarity, grammar, and style. No generative AI tools were used to generate research ideas, conduct experiments, analyze data, generate or modify results, or produce other research artifacts. The AI-assisted text was carefully reviewed and revised by the authors to ensure its accuracy, consistency with the underlying work, and appropriate attribution of claims. We take responsibility for the final content of this work, including all text, claims, and research artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work does not raise any specific ethical considerations under the topics outlined in the relevant Code of Ethics. It does not involve human subjects or sensitive personal data, and does not introduce privacy, security, discrimination, safety, legal, or research-integrity concerns beyond those ordinarily associated with scientific research. The authors have no additional ethical considerations to disclose.

### REPRODUCIBILITY STATEMENT

We provide the implementation, experimental configurations, and instructions required to reproduce the results in this paper. Our experiments generally use publicly available datasets, and we specify the data preprocessing, model architecture, training procedure, hyperparameters, evaluation metrics, and random seeds in the paper and supplementary material. Our code and configuration files will be made publicly available upon publication.

## REFERENCES

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. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Dangxing Chen and Weicheng Ye. How to address monotonicity for model risk management? In *International Conference on Machine Learning*, pp. 5282–5295. PMLR, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. TabTransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:67855860.

Hyunho Kim and Jong-Seok Lee. Scalable monotonic neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DjIsNDEOYX.

Benjamin J. Lengerich, S. Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional anova: An efficient algorithm for recovering identifiable additive models. In *International Conference on Artificial Intelligence and Statistics*, 2019. URL https://api.semanticscholar.org/CorpusID:207863661.

Xingchao Liu, Xing Han, Na Zhang, and Qiang Liu. Certified monotonic neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Niklas Nolte, Ouail Kitouni, and Mike Williams. Expressive monotonic neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=w2P7fMy_RH.

Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. *Advances in Neural Information Processing Systems*, 35:8414–8426, 2022.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Davor Runje and Sharath M Shankaranarayana. Constrained monotonic neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Davide Sartor, Alberto Sinigaglia, and Gian Antonio Susto. Advancing constrained monotonic neural networks: achieving universal approximation beyond bounded activations. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Joseph Sill. Monotonic networks. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pp. 661–667, Cambridge, MA, USA, 1998. MIT Press. ISBN 0262100762.

Aishwarya Sivaraman, Golnoosh Farnadi, Todd Millstein, and Guy Van den Broeck. Counterexample-guided learning of monotonic neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pp. 180–186, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375830. URL https://doi.org/10.1145/3375627.3375830.

Anton Frederik Thielmann, Arik Reuter, and Benjamin Säfken. Beyond black-box predictions: Identifying marginal feature effects in tabular transformer networks. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 1540–1548. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/thielmann26a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Zebin Yang, Aijun Zhang, and Agus Sudjianto. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, 120:108192, 2021.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations*, volume 2025, pp. 144–164, 2025.

Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya R. Gupta. Deep lattice networks and partial monotonic functions. *ArXiv*, abs/1709.06680, 2017. URL https://api.semanticscholar.org/CorpusID:25221722.

# A APPENDIX

The appendix is structured in the following way:

- Section A.1: We report further information about the datasets employed for this work.
- Section A.2: We report further information regarding how the results have been obtained

## A.1 DATASET DESCRIPTION

For this work, datasets employed in Sartor et al. (2025), Radenovic et al. (2022), Thielmann et al. (2026) have been selected. We also leverage an in-house generated Cloud Resource Management dataset. Next we provide a brief description of the employed datasets, but for a more in-depth explanation and more detailed description, refer to the original works and the links provided in Section 5.

- **AutoMPG:** This dataset is a regression dataset aimed at predicting the miles-per-gallon consumption and is comprised of 7 features, 3 of which are monotonic.
- **California Housing:** This dataset is a regression dataset aimed at predicting the median house value for California districts, expressed in hundreds of thousands of dollars. This dataset was derived from the 1990 U.S. census, using one row per census block group. It is comprised of 9 features, 1 of which is monotonic.
- **Diamond:** This dataset is a regression dataset containing observations of diamonds described by their carat weight, color, clarity, physical dimensions, and other attributes. Of the 10 features, we impose non-decreasing monotonic constraints on Carat, Color, Clarity, X, Y, and Z based on their expected relationship with price. Color and Clarity are encoded so that higher values indicate higher quality, while X, Y, and Z capture physical dimensions. These constraints reflect prior structural assumptions rather than strict empirical monotonicity, particularly because X, Y, and Z are correlated with carat and may capture differences in diamond proportions.
- **Cloud Management Data:** This dataset is a regression dataset composed of 7 features, 4 of which are monotonic non-decreasing, and the rest are monotonic non-increasing.
- **Heart Disease:** This dataset is a classification dataset composed of 13 features, 2 of which are monotonic, aimed at predicting a possible heart disease.
- **Loan Defaulter:** This dataset is a classification dataset composed of 28 features, 5 of which are monotonic, and is aimed at predicting load defaults.

## A.2 EXPERIMENTS DESCRIPTION

Following are the specifications used to obtain the results in Section 5. The experiments were developed in PyTorch and the code has been heavily based on the code provided by Sartor et al. (2025), Radenovic et al. (2022), Thielmann et al. (2026), and Ye et al. (2025). For all datasets, no hyperparameter tuning has been carried out.

Table 6: Hyperparameter and architectural configurations.

| Hyperparameter | MTAM | MLP | NBM | FT-Transformer | NAMformer |
|---|---|---|---|---|---|
| Learning rate | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| Number of epochs | $10^3$ | $10^3$ | $10^3$ | $10^3$ | $10^3$ |
| Hidden layer dimensions | [64, 32, 16, 16] | [64, 32, 16, 16] | [64, 32, 16, 16] | NA | [64, 32, 16, 16] |
| Dropout | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Feature Dropout | 0.3 | NA | 0.3 | NA | 0.3 |
| Embedding Dim | 32 | NA | NA | 32 | 32 |
| Transformer Depth | 2 | NA | NA | 2 | 2 |
| Number of Heads | 4 | NA | NA | 4 | 4 |
| Optimizer | Adam | Adam | Adam | Adam | Adam |
| Weight Decay | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ | $10^{-4}$ |
| Activation | ReLU | ReLU | ReLU | GeGLU | ReLU |
| Patience | 10 | 10 | 10 | 10 | 10 |