CONTEXT-CONDITIONED MONOTONIC RESPONSE SURFACE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In a partially monotonic setting, unconstrained features may not only affect the output nonmonotonically but also change the shape of the monotonic response. Additive branches and scalar calibration can limit context-dependent variation in the constrained response, while lattice architectures support richer interactions. To model this variation explicitly, Context-Conditioned Monotonic Response Surface Networks (CMRSN) use unconstrained context features to modulate only monotonicity-safe quantities, including positive basis scales, basis shifts, expert gates, and normalized nonnegative score directions. Each expert combines a context-modulated smooth shape block with an ordered Bernstein block to model multivariate monotonic response surfaces whose shapes vary across contexts. Theoretical analysis establishes exact partial monotonicity and, with sufficient model capacity, universal approximation of continuous partially monotonic score functions. Experiments on regression, binary classification, and ordinal classification show competitive predictive performance while maintaining zero sampled monotonicity violations. A controlled multivariate synthetic task further demonstrates recovery of context-dependent response geometry by comparing results with known conditional probabilities and expected-response surfaces.

1 INTRODUCTION

Monotonicity is a common structural prior in applications such as risk assessment, credit scoring, and medical decision support (Abu-Mostafa, 1990; Dugas et al., 2009; Gupta et al., 2016; Jeong et al., 2025; Wargnier-Dauchelle et al., 2025). However, in many applications, domain knowledge constrains only selected features. The remaining features are unconstrained and may both affect the prediction nonmonotonically and change the shape of the monotonic response.

For an input $\mathbf{x} = (\mathbf{x}_M, \mathbf{x}_U)$, partial monotonicity requires a scalar score to be nondecreasing in the constrained features $\mathbf{x}_M$ for every fixed $\mathbf{x}_U$, while imposing no monotonicity constraint with respect to the unconstrained features $\mathbf{x}_U$. Neural approaches address this requirement through structural constraints (Daniels & Velikova, 2010; Runje & Shankaranarayana, 2023; Kim & Lee, 2024; Sartor et al., 2025), monotonicity regularization (Gupta et al., 2021; Monteiro et al., 2022), or general property-enforcement frameworks (Francobaldi et al., 2026). Architectures such as deep lattice, hierarchical lattice, and scalable monotonic networks can also model interactions between constrained and unconstrained features (You et al., 2017; Yanagisawa et al., 2022; Kim & Lee, 2024). Hierarchical lattices are particularly relevant because unconstrained features can determine ordered lattice-vertex values. These developments motivate investigating how context can modulate monotonic basis geometry and aggregation while preserving partial monotonicity by construction.

Consider predicting concrete compressive strength. Under comparable curing conditions, strength generally increases with age for a fixed mixture, but the age–strength curve can differ across mixtures. Mixtures with little fly ash or blast-furnace slag may gain strength rapidly at early ages and then level off, whereas mixtures rich in supplementary cementitious materials may develop more slowly initially but continue gaining strength later. The predicted strength can thus be constrained to be nondecreasing in curing age, while slag, fly ash, and aggregate variables provide context that may alter the response's slope, curvature, and saturation. Accordingly, the mixture context should condition the response shape rather than enter only as an additive offset.

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.

Context-Conditioned Monotonic Response Surface Networks (CMRSN) address this problem by treating $\mathbf{x}_U$ as context for a family of monotonic response surfaces over $\mathbf{x}_M$. A context encoder generates soft gates over monotonic experts, positive scales and unconstrained shifts for smooth monotonic bases, and normalized nonnegative score directions. For fixed $\mathbf{x}_U$, these quantities are constant with respect to $\mathbf{x}_M$ and enter through order-preserving operations, allowing context-dependent response shapes while preserving partial monotonicity of the resulting score.

The main contributions are:

- CMRSN conditions expert gates, positive basis scales, basis shifts, and normalized nonnegative score directions on $\mathbf{x}_U$. These monotonicity-safe modulation mechanisms allow the response over $\mathbf{x}_M$ to vary with context while preserving partial monotonicity.
- A unified monotonic expert dictionary combines context-modulated smooth shape components with an ordered Bernstein component for expressive and trainable monotonic response-surface modeling.
- Theoretical analysis establishes exact partial monotonicity and universal approximation of continuous partially monotonic score functions by the CMRSN model family with sufficient capacity.

2 Related Work

Monotonic and partially monotonic neural networks. Early work introduced monotonicity as a learning hint or architectural constraint in neural prediction (Abu-Mostafa, 1990; Archer & Wang, 1993), and Daniels & Velikova (2010) explicitly studied monotonic and partially monotonic neural networks. Recent constrained architectures improve expressiveness and scalability (Runje & Shankaranarayana, 2023; Kim & Lee, 2024; Sartor et al., 2025), while general property-enforcement frameworks provide complementary guarantees over the input domain (Francobaldi et al., 2026). Other approaches include calibrated lookup tables and lattice models (Gupta et al., 2016; Milani Fard et al., 2016; You et al., 2017; Bakst et al., 2021; Yanagisawa et al., 2022), integral representations (Wehenkel & Louppe, 2019), general neural-network verification (Bunel et al., 2018), monotonicity-specific certification or counterexample-guided learning (Liu et al., 2020; Sivaraman et al., 2020; Polo-Molina et al., 2026a), Lipschitz-constrained residuals (Nolte et al., 2023), smooth min-max modules (Igel, 2024), and monotonic spline-based KANs (Polo-Molina et al., 2026b). Several of these architectures support monotonicity constraints on selected coordinates and interactions between constrained and unconstrained inputs. In particular, the hierarchical lattice layer (HLL) conditions ordered lattice-vertex values on unconstrained inputs, while the Kronecker-factored lattice (KFL) factorizes lattice parameters to control representation cost (Yanagisawa et al., 2022; Bakst et al., 2021). CMRSN instead combines context-modulated smooth bases with context-independent ordered Bernstein coefficients within each expert, and conditions the normalized coordinate and expert weights on $\mathbf{x}_U$.

Isotonic embedding and calibration. Deep isotonic embedding networks combine monotonic feature embeddings with nonnegative weights derived from unconstrained inputs and a monotonic feature-learning module (Zhao et al., 2024). Normalization-aware isotonic methods instead address multiclass probability calibration (Arad & Rosset, 2025). CMRSN combines context-modulated smooth bases and ordered multivariate Bernstein components within an expert dictionary, with context-dependent normalized aggregation within and across experts.

3 Method

3.1 Problem setup

Let $\mathbf{x} = (\mathbf{x}_M, \mathbf{x}_U)$, where $\mathbf{x}_M \in K_M \subset \mathbb{R}^{d_M}$ contains monotonicity-constrained features and $\mathbf{x}_U \in \mathbb{R}^{d_U}$ contains unconstrained context features. We model the constrained inputs on a compact rectangle $K_M = \prod_{j=1}^{d_M} [\underline{x}_{M,j}, \overline{x}_{M,j}]$, with $\underline{x}_{M,j} < \overline{x}_{M,j}$ for every coordinate. Features constrained to be nonincreasing are sign-aligned during preprocessing, so every coordinate of $\mathbf{x}_M$ is treated as nondecreasing. A scalar score function $f : K_M \times \mathbb{R}^{d_U} \to \mathbb{R}$ is partially monotonic with respect to

$\mathbf{x}_M$ if, for every fixed $\mathbf{x}_U$ and all $\mathbf{x}_M, \mathbf{x}'_M \in K_M$,

$$\mathbf{x}_M \preceq \mathbf{x}'_M \quad \Longrightarrow \quad f(\mathbf{x}_M, \mathbf{x}_U) \le f(\mathbf{x}'_M, \mathbf{x}_U), \tag{1}$$

where $\preceq$ denotes component-wise order on K_M; no ordering is required across distinct contexts.

CMRSN maps the unconstrained features to a context representation

$$\mathbf{c} = q_\varphi(\mathbf{x}_U) \in \mathbb{R}^{d_c}, \tag{2}$$

where q_φ is an unrestricted neural context encoder with parameters φ.

3.2 Context-conditioned monotonic expert dictionary

Context modulation. For a smooth block with L layers, let m_ℓ and n_ℓ denote the preactivation and output widths of layer ℓ, respectively, and let J_ℓ denote its number of basis replicas. The context vector is passed to separate gating and basis-modulation heads. The gating head defines a soft selection over E monotonic experts:

$$\pi_e(\mathbf{c}) = \frac{\exp(\mathbf{w}_{\pi,e}^\top \mathbf{c} + b_{\pi,e})}{\sum_{e'=1}^{E} \exp(\mathbf{w}_{\pi,e'}^\top \mathbf{c} + b_{\pi,e'})}, \qquad e = 1, \dots, E, \tag{3}$$

where $\mathbf{w}_{\pi,e} \in \mathbb{R}^{d_c}$ and $b_{\pi,e} \in \mathbb{R}$ are learnable parameters, $\pi_e(\mathbf{c}) \ge 0$, and $\sum_{e=1}^{E} \pi_e(\mathbf{c}) = 1$. For layer $\ell = 1, \dots, L$, expert $e = 1, \dots, E$, replica $r = 1, \dots, J_\ell$, and basis type $t \in \{+, -, s\}$, define

$$\begin{aligned} \boldsymbol{\kappa}^t_{\ell,e,r}(\mathbf{c}) &= \operatorname{softplus}\big(\bar{\boldsymbol{\kappa}}^t_{\ell,e,r} + A^t_{\kappa,\ell,e,r}\mathbf{c}\big)\,, \\ \boldsymbol{\tau}^t_{\ell,e,r}(\mathbf{c}) &= \bar{\boldsymbol{\tau}}^t_{\ell,e,r} + A^t_{\tau,\ell,e,r}\mathbf{c}. \end{aligned} \tag{4}$$

The vectors $\bar{\boldsymbol{\kappa}}^t_{\ell,e,r}, \bar{\boldsymbol{\tau}}^t_{\ell,e,r} \in \mathbb{R}^{m_\ell}$ are context-independent base vectors; $A^t_{\kappa,\ell,e,r}, A^t_{\tau,\ell,e,r} \in \mathbb{R}^{m_\ell \times d_c}$ are learnable modulation matrices. Thus $\boldsymbol{\kappa}^t_{\ell,e,r}(\mathbf{c}), \boldsymbol{\tau}^t_{\ell,e,r}(\mathbf{c}) \in \mathbb{R}^{m_\ell}$. The scales are strictly positive, whereas the shifts remain unconstrained because translating a nondecreasing basis does not change its monotonicity. Eq. (3) operates across experts, while Eq. (4) operates inside each expert; the two heads share the context representation but have distinct parameters.

Smooth monotonic component. Each expert maps $\mathbf{x}_M$ to a monotonic representation under the current context. Let $\mathbf{h}^{(0)}_e = \mathbf{x}_M \in \mathbb{R}^{d_M}$ and set $n_0 = d_M$ for dimensional consistency across layers. At layer ℓ, the preactivation of expert e is

$$\mathbf{a}^{(\ell)}_e = W^{(\ell)}_{a,e}\mathbf{h}^{(\ell-1)}_e + \mathbf{b}^{(\ell)}_{a,e}, \tag{5}$$

where $W^{(\ell)}_{a,e} \in \mathbb{R}^{m_\ell \times n_{\ell-1}}_+$ is a learnable nonnegative weight matrix and $\mathbf{b}^{(\ell)}_{a,e} \in \mathbb{R}^{m_\ell}$ is an unconstrained bias vector. Consequently, $\mathbf{a}^{(\ell)}_e \in \mathbb{R}^{m_\ell}$. For any $\mathbf{a} \in \mathbb{R}^{m_\ell}$, the convex, concave, and saturated basis functions are

$$\begin{aligned} \boldsymbol{\phi}^+_{\ell,e,r}(\mathbf{a};\mathbf{c}) &= \operatorname{softplus}\Big(\boldsymbol{\kappa}^+_{\ell,e,r}(\mathbf{c}) \odot \big(\mathbf{a} - \boldsymbol{\tau}^+_{\ell,e,r}(\mathbf{c})\big)\Big) \oslash \boldsymbol{\kappa}^+_{\ell,e,r}(\mathbf{c}), \\ \boldsymbol{\phi}^-_{\ell,e,r}(\mathbf{a};\mathbf{c}) &= -\operatorname{softplus}\Big(-\boldsymbol{\kappa}^-_{\ell,e,r}(\mathbf{c}) \odot \big(\mathbf{a} - \boldsymbol{\tau}^-_{\ell,e,r}(\mathbf{c})\big)\Big) \oslash \boldsymbol{\kappa}^-_{\ell,e,r}(\mathbf{c}), \\ \boldsymbol{\phi}^s_{\ell,e,r}(\mathbf{a};\mathbf{c}) &= \operatorname{sigmoid}\big(\boldsymbol{\kappa}^s_{\ell,e,r}(\mathbf{c}) \odot \big(\mathbf{a} - \boldsymbol{\tau}^s_{\ell,e,r}(\mathbf{c})\big)\big)\,. \end{aligned} \tag{6}$$

Here $\odot$ and $\oslash$ denote element-wise multiplication and division, respectively. Concatenating the basis responses gives

$$\Phi_{\ell,e}(\mathbf{a};\mathbf{c}) = \Big[\{\boldsymbol{\phi}^+_{\ell,e,r}(\mathbf{a};\mathbf{c})\}_{r=1}^{J_\ell}; \{\boldsymbol{\phi}^-_{\ell,e,r}(\mathbf{a};\mathbf{c})\}_{r=1}^{J_\ell}; \{\boldsymbol{\phi}^s_{\ell,e,r}(\mathbf{a};\mathbf{c})\}_{r=1}^{J_\ell}\Big] \in \mathbb{R}^{3J_\ell m_\ell}. \tag{7}$$

The resulting representation layer is

$$\mathbf{h}^{(\ell)}_e = W^{(\ell)}_{o,e}\Phi_{\ell,e}\Big(\mathbf{a}^{(\ell)}_e;\mathbf{c}\Big) + R^{(\ell)}_e\mathbf{h}^{(\ell-1)}_e + \mathbf{b}^{(\ell)}_{o,e}, \tag{8}$$

where $\mathbf{h}^{(\ell)}_e \in \mathbb{R}^{n_\ell}$, $W^{(\ell)}_{o,e} \in \mathbb{R}^{n_\ell \times 3J_\ell m_\ell}_+$, $R^{(\ell)}_e \in \mathbb{R}^{n_\ell \times n_{\ell-1}}_+$, and $\mathbf{b}^{(\ell)}_{o,e} \in \mathbb{R}^{n_\ell}$. The bias is unconstrained, while the nonnegative residual map preserves monotonicity and facilitates optimization. For fixed $\mathbf{c}$, all three basis families are coordinate-wise nondecreasing in $\mathbf{a}$. Together with the nonnegative weights in the preactivation, output, and residual maps, this makes every $\mathbf{h}^{(\ell)}_e$ coordinate-wise nondecreasing in $\mathbf{x}_M$. The smooth component output is $\mathbf{z}^S_e(\mathbf{x}_M;\mathbf{c}) = \mathbf{h}^{(L)}_e \in \mathbb{R}^{n_L}$.

Ordered Bernstein component. Let $T_M : K_M \to [0,1]^{d_M}$ be the fixed coordinate-wise affine rescaling $\tilde{x}_{M,j} = (x_{M,j} - \underline{x}_{M,j})/(\overline{x}_{M,j} - \underline{x}_{M,j})$, where $\underline{x}_{M,j} < \overline{x}_{M,j}$ are the coordinate bounds of K_M, and write $\widetilde{\mathbf{x}}_M = T_M(\mathbf{x}_M)$. The positive scaling factors ensure that T_M preserves componentwise order. Each expert includes a tensor-product Bernstein polynomial of degree N in each constrained coordinate:

$$z_e^B(\mathbf{x}_M) = \sum_{\mathbf{k}\in\{0,\dots,N\}^{d_M}} \theta_{e,\mathbf{k}}^B \prod_{j=1}^{d_M} \binom{N}{k_j} \tilde{x}_{M,j}^{k_j} (1-\tilde{x}_{M,j})^{N-k_j}. \tag{9}$$

Here $\mathbf{k} = (k_1, \dots, k_{d_M})$ is a multi-index and $\theta_{e,\mathbf{k}}^B \in \mathbb{R}$ is its associated coefficient. Let $\boldsymbol{\theta}_e^B$ collect all $(N+1)^{d_M}$ coefficients of expert e. Monotonicity is enforced by requiring $\boldsymbol{\theta}_e^B \in \mathcal{C}_B$, where

$$\mathcal{C}_B = \left\{ \boldsymbol{\zeta} \in \mathbb{R}^{(N+1)^{d_M}} : \zeta_{\mathbf{k}} \le \zeta_{\mathbf{k}+\mathbf{e}_j},\ \forall j = 1,\dots,d_M,\ \mathbf{k} \in \{0,\dots,N\}^{d_M} \text{ with } k_j < N \right\}, \tag{10}$$

where $\mathbf{e}_j$ is the jth standard basis vector in the multi-index space. Because T_M is coordinate-wise increasing, ordered coefficients make z_e^B nondecreasing in $\mathbf{x}_M$ on K_M. After each Adam update produces tentative coefficients $\bar{\boldsymbol{\theta}}_e^B$, the cone-enforcing isotonic repair operator $\mathcal{R}_{\mathcal{C}_B}$ is applied:

$$\boldsymbol{\theta}_e^B \leftarrow \mathcal{R}_{\mathcal{C}_B}\left(\bar{\boldsymbol{\theta}}_e^B\right), \qquad \mathcal{R}_{\mathcal{C}_B}\left(\bar{\boldsymbol{\theta}}_e^B\right) \in \mathcal{C}_B. \tag{11}$$

For each expert, the implementation applies cumulative maxima along every tensor axis. Equivalently,

$$[\mathcal{R}_{\mathcal{C}_B}(\bar{\boldsymbol{\theta}})]_{\mathbf{k}} = \max_{\mathbf{j}\preceq\mathbf{k}} \bar{\theta}_{\mathbf{j}}. \tag{12}$$

The predecessor sets are nested under coordinate-wise order, so the repaired coefficients belong to $\mathcal{C}_B$. This is an order-enforcing repair, not a nearest-point Euclidean projection. Degree N controls capacity, and the coefficients are trained jointly with the smooth component. The tensor-product Bernstein block contains $E(N+1)^{d_M}$ coefficients, and one cumulative-maximum sweep costs $\mathcal{O}(Ed_M(N+1)^{d_M})$. Both costs grow exponentially with the constrained dimension, limiting the scalability of the full tensor-product construction.

Unified expert representation. The smooth component provides context-dependent shape variation, whereas the ordered Bernstein component supplies a monotonic polynomial basis used in the approximation analysis. Their concatenation defines the expert representation

$$g_{\theta_e}(\mathbf{x}_M; \mathbf{c}) = \left[\mathbf{z}_e^S(\mathbf{x}_M; \mathbf{c}),\ z_e^B(\mathbf{x}_M)\right] \in \mathbb{R}^{n_L+1}. \tag{13}$$

Here θ_e collects the trainable parameters of expert e. To aggregate the $n_L + 1$ expert coordinates without reversing monotonicity, a separate context head defines the nonnegative score direction

$$\mathbf{v}_e(\mathbf{c}) = \mathrm{softmax}(\bar{\mathbf{v}}_e + A_{v,e}\mathbf{c}), \qquad \sum_{j=1}^{n_L+1} v_{e,j}(\mathbf{c}) = 1, \tag{14}$$

where $\bar{\mathbf{v}}_e \in \mathbb{R}^{n_L+1}$ and $A_{v,e} \in \mathbb{R}^{(n_L+1)\times d_c}$ are learnable and $v_{e,j}(\mathbf{c}) > 0$. The entries $j = 1,\dots,n_L$ weight the smooth coordinates, and entry $j = n_L + 1$ weights the Bernstein coordinate. Normalization separates within-expert coordinate weights from the expert gates: $\sum_j \pi_e(\mathbf{c}) v_{e,j}(\mathbf{c}) = \pi_e(\mathbf{c})$.

3.3 Prediction and training

We instantiate CMRSN for regression, binary classification, and ordinal classification. Given the training set $\mathcal{D}_{\mathrm{tr}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\mathrm{tr}}}$, the model first maps each input to the scalar score

$$s_\Theta(\mathbf{x}) = b_\psi(\mathbf{c}) + \sum_{e=1}^{E} \pi_e(\mathbf{c}) \mathbf{v}_e(\mathbf{c})^\top g_{\theta_e}(\mathbf{x}_M; \mathbf{c}), \tag{15}$$

where $b_\psi(\mathbf{c}) \in \mathbb{R}$ is an unrestricted context-dependent bias with parameters ψ, and Θ collects the context encoder, expert, modulation, aggregation, bias, and task-specific head parameters.

For regression, targets are standardized using the training mean μ_y and standard deviation $\sigma_y > 0$. The score predicts the standardized target, and the reported prediction is $\hat{y} = \mu_y + \sigma_y s_\Theta(\mathbf{x})$.

For binary labels $y \in \{0, 1\}$, the positive-class probability is $\mathrm{sigmoid}(s_\Theta(\mathbf{x}))$ and the predicted label uses a probability threshold of 0.5. Both mappings preserve monotonicity in the score.

For ordinal labels $y \in \{1, \dots, C\}$, ordered thresholds map the score to the predicted label:

$$\hat{y}(\mathbf{x}) = 1 + \sum_{k=1}^{C-1} \mathbb{I}[s_\Theta(\mathbf{x}) > \vartheta_k], \qquad \vartheta_1 < \cdots < \vartheta_{C-1}, \tag{16}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The threshold ordering is enforced by the reparameterization $\vartheta_1 = \widetilde{\vartheta}_1$ and $\vartheta_k = \vartheta_{k-1} + \mathrm{softplus}(\widetilde{\Delta}_k)$ for $k = 2, \dots, C-1$, where $\widetilde{\vartheta}_1$ and $\widetilde{\Delta}_k$ are unconstrained scalar parameters. The corresponding cumulative ordinal probabilities are

$$P_\Theta(y > k \mid \mathbf{x}) = \mathrm{sigmoid}(s_\Theta(\mathbf{x}) - \vartheta_k), \qquad k = 1, \dots, C-1. \tag{17}$$

Both the threshold prediction and the cumulative probabilities are nondecreasing in s_Θ; hence, the ordinal head preserves partial monotonicity.

Smooth-layer weights required to be nonnegative are softplus-reparameterized. In particular,

$$W_{a,e}^{(\ell)} = \mathrm{softplus}(\widetilde{W}_{a,e}^{(\ell)}), \qquad W_{o,e}^{(\ell)} = \mathrm{softplus}(\widetilde{W}_{o,e}^{(\ell)}), \qquad R_e^{(\ell)} = \mathrm{softplus}(\widetilde{R}_e^{(\ell)}). \tag{18}$$

The tilded matrices are unconstrained raw parameters, and $\mathrm{softplus}$ is applied element-wise. The basis scales and normalized score directions are constrained by Eqs. (4) and (14), and the Bernstein coefficients satisfy Eq. (10). Remaining raw parameters, including the context encoder, gate logits, smooth-layer biases, basis shifts, context bias, and modulation matrices, remain unconstrained.

Let $\mathrm{BCE}(z, p) = -z \log p - (1 - z) \log(1 - p)$ denote binary cross-entropy for a target $z \in \{0, 1\}$ and probability $p \in (0, 1)$. For regression, define the standardized target $\tilde{y}_i = (y_i - \mu_y)/\sigma_y$. For ordinal classification, define the cumulative binary targets $\xi_{ik} = \mathbb{I}[y_i > k]$ and predictions $p_{ik} = P_\Theta(y > k \mid \mathbf{x}_i)$ for $k = 1, \dots, C-1$. Binary classification uses the original labels $y_i \in \{0, 1\}$. The corresponding per-example losses are

$$\ell_i = \begin{cases} (s_\Theta(\mathbf{x}_i) - \tilde{y}_i)^2, & \text{regression,} \\ \mathrm{BCE}(y_i, \mathrm{sigmoid}(s_\Theta(\mathbf{x}_i))), & \text{binary,} \\ \dfrac{1}{C-1} \sum_{k=1}^{C-1} \mathrm{BCE}(\xi_{ik}, p_{ik}), & \text{ordinal.} \end{cases} \tag{19}$$

CMRSN minimizes

$$\min_\Theta \frac{1}{n_{\mathrm{tr}}} \sum_i \ell_i + \frac{\lambda_{\mathrm{wd}}}{2} \|\Theta\|_2^2, \tag{20}$$

where $\lambda_{\mathrm{wd}} \geq 0$ denotes the weight-decay coefficient, applied to the trainable raw variables. The model is trained end-to-end by Adam with weight decay, followed by Bernstein repair after each update.

Algorithm 1 summarizes the optimization procedure, including the context-conditioned forward pass, Adam updates, and isotonic repair of the Bernstein coefficients after each minibatch.

Theoretical guarantees. CMRSN is exactly partially monotonic because, for fixed $\mathbf{x}_U$, all context-dependent quantities are constant in $\mathbf{x}_M$ and every operation on the constrained pathway preserves coordinate-wise order. Nondecreasing preprocessing and the task-specific prediction rules preserve this property. Moreover, the ordered Bernstein experts and context-dependent softmax gates yield universal approximation of continuous partially monotonic scores on compact domains as capacity grows. Proofs are given in Appendices A.1–A.3.

4 Experiments

4.1 Experimental Setup

Real-data tasks. We evaluate eight real-data prediction tasks: regression on BlogFeedback, Concrete Strength, and Auto MPG; binary classification on COMPAS and Adult; and ordinal classification on Wine Quality, Car Evaluation, and Maternal Health Risk. Dataset statistics and feature partitions are listed in Table 4 of Appendix B.1.

Algorithm 1 Training CMRSN

Input: Training set $\mathcal{D}_{\mathrm{tr}}$; task type; expert count E; Bernstein degree N; number of epochs T; optimizer settings

1: Initialize the trainable parameters and, for ordinal classification, raw threshold variables; construct the ordered thresholds when applicable and repair the Bernstein tensors by Eq. (12).
2: **for** $t = 1, \ldots, T$ **do**
3: **for** each minibatch $\mathcal{B} \subset \mathcal{D}_{\mathrm{tr}}$ **do**
4: Encode $\mathbf{c}_i = q_\varphi(\mathbf{x}_U^{(i)})$ and compute the expert gates, basis modulation, and normalized nonnegative score directions using Eqs. (3), (4), and (14).
5: Evaluate the smooth and Bernstein components, concatenate them into $g_{\theta_e}(\mathbf{x}_M^{(i)}; \mathbf{c}_i)$, and aggregate the score using Eq. (15).
6: Compute the task-specific predictions and minibatch objective using Eqs. (19) and (20).
7: Update all trainable parameters by Adam with weight decay.
8: Repair each Bernstein coefficient tensor into the isotonic cone using the coordinate-wise cumulative-maximum operation in Eq. (12).

Output: Trained parameters $\widehat{\Theta}$, including ordered thresholds for ordinal classification

Baselines. We compare CMRSN with seven baselines: an unconstrained multilayer perceptron (MLP) (Hornik et al., 1989), a weight-constrained MLP (Wc-MLP) (Sill, 1997), an additive partially monotonic neural network (A-PMNN) (Daniels & Velikova, 2010; Trindade et al., 2021), a smooth min-max monotonic network (SMM) (Igel, 2024), a constrained monotonic neural network (CoMNN) (Runje & Shankaranarayana, 2023), a hierarchical lattice layer (HLL) (Yanagisawa et al., 2022), and a Kronecker-factored lattice (KFL) (Bakst et al., 2021). All methods produce a scalar score and use the same task-specific prediction protocol. Implementations are summarized in Table 5 in Appendix B.2, and selected parameter counts are reported in Table 12 in Appendix C.3.

Evaluation metrics. Regression uses normalized root mean squared error (NRMSE) and normalized mean absolute error (NMAE), both divided by the training-target standard deviation. Binary classification uses area under the receiver operating characteristic curve (AUROC) and accuracy (Acc.); ordinal classification uses Acc., label MAE divided by $C - 1$ (NMAE), and quadratic weighted kappa (QWK). Across tasks, empirical monotonicity violation rate (MVR) is

$$\mathrm{MVR} = \frac{1}{N_{\mathrm{pair}}} \sum_{i=1}^{N_{\mathrm{pair}}} \mathbb{I}\Big[s_\Theta(\mathbf{x}_M'^{(i)}, \mathbf{x}_U^{(i)}) < s_\Theta(\mathbf{x}_M^{(i)}, \mathbf{x}_U^{(i)}) - \epsilon_{\mathrm{mvr}} \Big], \tag{21}$$

for N_{pair} sampled pairs with identical context and $\mathbf{x}_M'^{(i)} \succeq \mathbf{x}_M^{(i)}$ after sign alignment, using numerical tolerance $\epsilon_{\mathrm{mvr}} > 0$.

Training protocol. Six datasets use outer five-fold evaluation; BlogFeedback and Adult each share their official test set across five repeated training–validation splits. Within each split, tuning uses seed 42 and a fixed inner validation subset comprising approximately 20% of the training pool; the test set is reserved for final evaluation. Selected configurations are retrained with seeds 43–47. Means $\pm$ standard deviations are computed across seeds after averaging splits within datasets and, for task-level summaries, equally weighting datasets. All methods share training-only preprocessing, sign alignment, and task-specific losses and prediction rules. CMRSN fixes $E = 4, N = 2$. In Tables 1–3, bold and underlining indicate the best and second-best predictive or recovery means, respectively; MVR is not highlighted.

4.2 Comparison Results and Analysis

Table 1 summarizes the macro-averaged comparison within each task. The unconstrained MLP remains strong in regression and ordinal prediction but retains nonzero MVR across all three tasks. Among the methods with zero sampled MVR, CMRSN attains the lowest mean regression NRMSE and NMAE, while CoMNN attains the highest mean binary AUROC and Acc. For ordinal prediction, CoMNN leads Acc. and NMAE among the zero-MVR methods, whereas KFL leads QWK. Complete per-dataset results are reported in Tables 6–8 in Appendix C.1.

Table 1: Task-level comparison. Values are mean ± standard deviation across five seeds, after averaging splits within datasets and datasets within tasks.

	Regression			Binary		
Method	NRMSE ↓	NMAE ↓	MVR ↓	AUROC ↑	Acc. ↑	MVR ↓
MLP	**0.4661 ± 0.0055**	**0.2168 ± 0.0030**	0.1559 ± 0.0221	0.8199 ± 0.0004	0.7669 ± 0.0014	0.1111 ± 0.0401
Wc-MLP	0.5822 ± 0.0004	0.3347 ± 0.0019	0.0000	0.7956 ± 0.0001	0.7513 ± 0.0001	0.0000
A-PMNN	0.5667 ± 0.0048	0.3151 ± 0.0017	0.0000	0.8165 ± 0.0004	0.7640 ± 0.0006	0.0000
SMM	0.5096 ± 0.0036	0.2649 ± 0.0016	0.0000	0.8180 ± 0.0005	0.7647 ± 0.0010	0.0000
CoMNN	0.5219 ± 0.0154	0.2757 ± 0.0127	0.0000	**0.8218 ± 0.0003**	**0.7689 ± 0.0008**	0.0000
HLL	0.6756 ± 0.0035	0.4128 ± 0.0027	0.0000	0.8024 ± 0.0015	0.7332 ± 0.0037	0.0000
KFL	0.5828 ± 0.0616	0.2797 ± 0.0032	0.0000	0.8189 ± 0.0001	0.7684 ± 0.0006	0.0000
CMRSN	0.4955 ± 0.0071	0.2501 ± 0.0040	0.0000	0.8202 ± 0.0008	0.7676 ± 0.0009	0.0000

Ordinal

Method	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
MLP	**0.7401 ± 0.0056**	**0.0942 ± 0.0023**	**0.7348 ± 0.0062**	0.2383 ± 0.0126
Wc-MLP	0.6468 ± 0.0029	0.1305 ± 0.0015	0.6198 ± 0.0013	0.0000
A-PMNN	0.6513 ± 0.0027	0.1279 ± 0.0011	0.6317 ± 0.0038	0.0000
SMM	0.6840 ± 0.0017	0.1163 ± 0.0008	0.6482 ± 0.0034	0.0000
CoMNN	0.6970 ± 0.0026	0.1113 ± 0.0009	0.6817 ± 0.0013	0.0000
HLL	0.5925 ± 0.0044	0.1613 ± 0.0012	0.2977 ± 0.0028	0.0000
KFL	0.6799 ± 0.0027	0.1120 ± 0.0009	0.6997 ± 0.0029	0.0000
CMRSN	0.6832 ± 0.0028	0.1153 ± 0.0008	0.6582 ± 0.0046	0.0000

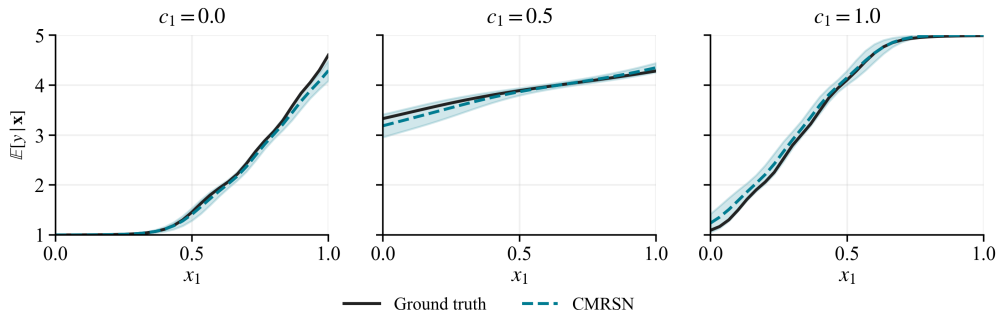

Figure 1: Response cross-sections at $x_2 = c_2 = 0.5$, $z = 0$. Panels vary c_1. Solid: ground truth; dashed: CMRSN mean; shading: one standard deviation over five seeds.

4.3 Ablation Study

Table 2 reports the CMRSN ablation results under the selected full-model configurations, without separately retuning the variants. Replacing the context-dependent expert gates, basis scales and shifts, and score directions with global counterparts while retaining the context bias (*w/o Surface Conditioning*) produces the largest degradation in every task-level predictive mean except ordinal QWK. Removing the smooth-shape component and retaining the ordered Bernstein block (*w/o Smooth Block*) also worsens all reported predictive means and yields the largest drop in ordinal QWK, supporting the utility of smooth shape adaptation. Removing the ordered Bernstein component and retaining only the smooth-shape block (*w/o Bernstein Block*) has task-dependent effects: all three ordinal metrics worsen, whereas regression NRMSE and NMAE and binary AUROC and Acc. improve slightly. Replacing the context-dependent basis scales and shifts with global parameters (*w/o Basis Modulation*) degrades all reported predictive means, particularly for regression and ordinal classification. Full CMRSN achieves the best mean ordinal Acc., NMAE, and QWK among these variants. All variants retain zero sampled MVR; the differences reflect component utility under the fixed configurations, with capacity and optimization effects potentially contributing. Complete per-dataset ablation results are reported in Tables 9–11 in Appendix C.2.

Table 2: Ablation under the selected full-model configurations. Values are mean ± standard deviation across five seeds, after averaging splits within datasets and datasets within tasks.

	Regression			Binary		
Variant	NRMSE ↓	NMAE ↓	MVR ↓	AUROC ↑	Acc. ↑	MVR ↓
Full CMRSN	0.4955 ± 0.0071	0.2501 ± 0.0040	0.0000	0.8202 ± 0.0008	0.7676 ± 0.0009	0.0000
w/o Surface Conditioning	0.5517 ± 0.0024	0.3022 ± 0.0017	0.0000	0.8190 ± 0.0004	0.7646 ± 0.0008	0.0000
w/o Basis Modulation	0.5226 ± 0.0044	0.2749 ± 0.0058	0.0000	0.8197 ± 0.0007	0.7664 ± 0.0013	0.0000
w/o Bernstein Block	**0.4950 ± 0.0074**	**0.2462 ± 0.0048**	0.0000	**0.8210 ± 0.0006**	**0.7677 ± 0.0012**	0.0000
w/o Smooth Block	0.5171 ± 0.0041	0.2676 ± 0.0046	0.0000	0.8192 ± 0.0005	0.7659 ± 0.0006	0.0000

	Ordinal			
Variant	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
Full CMRSN	**0.6832 ± 0.0028**	**0.1153 ± 0.0008**	**0.6582 ± 0.0046**	0.0000
w/o Surface Conditioning	0.6539 ± 0.0073	0.1267 ± 0.0030	0.6224 ± 0.0066	0.0000
w/o Basis Modulation	0.6619 ± 0.0042	0.1240 ± 0.0018	0.6265 ± 0.0044	0.0000
w/o Bernstein Block	0.6788 ± 0.0031	0.1169 ± 0.0014	0.6553 ± 0.0040	0.0000
w/o Smooth Block	0.6764 ± 0.0019	0.1183 ± 0.0006	0.6143 ± 0.0026	0.0000

4.4 Synthetic Mechanism Validation

To evaluate recovery against known response surfaces, we construct an ordinal task with $\mathbf{x}_M = (x_1, x_2) \in [0,1]^2$ and context $\mathbf{x}_U = (c_1, c_2, z)$, where $c_1, c_2 \in [0,1]$ and $z \in \{0,1\}$. Context modulates the mixing weights and shape parameters of convex, concave, and S-shaped saturated functions of distinct positive projections of (x_1, x_2). The generator and five-class label model are specified in Appendix B.3.

We measure cumulative-probability RMSE (CP-RMSE) against known probabilities and response-surface MAE (Surf. MAE) on fixed context grids. Plots show the expected ordinal label

$$\mathbb{E}[y \mid \mathbf{x}] = 1 + \sum_{k=1}^{C-1} P(y > k \mid \mathbf{x}). \tag{22}$$

CP-RMSE measures cumulative-probability errors across ordinal thresholds, whereas Surf. MAE measures expected-label surface error. Errors across thresholds can cancel in the expectation, so the two metrics are complementary.

Table 3: Prediction and response recovery on the coupled shape-regime task. Values are mean ± standard deviation over five independent final seeds.

Method	Acc. ↑	NMAE ↓	QWK ↑	CP-RMSE ↓	Surf. MAE ↓	MVR ↓
MLP	0.7370 ± 0.0046	0.0669 ± 0.0013	0.9312 ± 0.0017	0.0629 ± 0.0054	0.1074 ± 0.0238	0.0012 ± 0.0008
Wc-MLP	0.4992 ± 0.0060	0.1497 ± 0.0022	0.7875 ± 0.0041	0.2475 ± 0.0011	0.6145 ± 0.0104	0.0000
A-PMNN	0.6097 ± 0.0064	0.1048 ± 0.0018	0.8783 ± 0.0027	0.1754 ± 0.0019	0.4526 ± 0.0030	0.0000
SMM	0.6575 ± 0.0017	0.0890 ± 0.0006	0.9025 ± 0.0018	0.1397 ± 0.0017	0.3474 ± 0.0073	0.0000
CoMNN	0.7383 ± 0.0037	0.0667 ± 0.0009	0.9310 ± 0.0011	0.0635 ± 0.0087	0.1127 ± 0.0300	0.0000
HLL	0.7191 ± 0.0057	0.0720 ± 0.0016	0.9266 ± 0.0024	0.0910 ± 0.0013	0.1607 ± 0.0311	0.0000
KFL	0.7418 ± 0.0029	0.0657 ± 0.0008	0.9319 ± 0.0009	0.0606 ± 0.0019	0.1218 ± 0.0122	0.0000
CMRSN	**0.7425 ± 0.0037**	**0.0655 ± 0.0010**	**0.9324 ± 0.0008**	**0.0577 ± 0.0054**	**0.0977 ± 0.0302**	0.0000

Quantitative comparison. Table 3 reports five-seed means and standard deviations. CMRSN has the highest mean Acc. and QWK and lowest mean NMAE, CP-RMSE, and Surf. MAE, with zero sampled MVR. KFL closely matches its ordinal prediction and CP-RMSE; CoMNN and MLP also have low recovery errors, although MLP retains nonzero sampled MVR. HLL has intermediate errors, while A-PMNN, SMM, and Wc-MLP have larger probability and surface discrepancies.

Geometry of the recovered response surfaces. In Fig. 2, increasing c_1 switches the generating score from an x_1-weighted convex surface through an x_2-weighted concave surface to diagonal S-shaped saturation. CMRSN recovers the changing orientation, slope, and plateau while remaining coordinate-wise monotonic; a context-dependent vertical offset alone cannot explain this variation.

Fig. 1 shows recovery of the sharp rise and plateau at $c_1 = 1$, but also underestimation near $x_1 = 1$ at $c_1 = 0$ and $x_1 = 0$ at $c_1 = 0.5$.

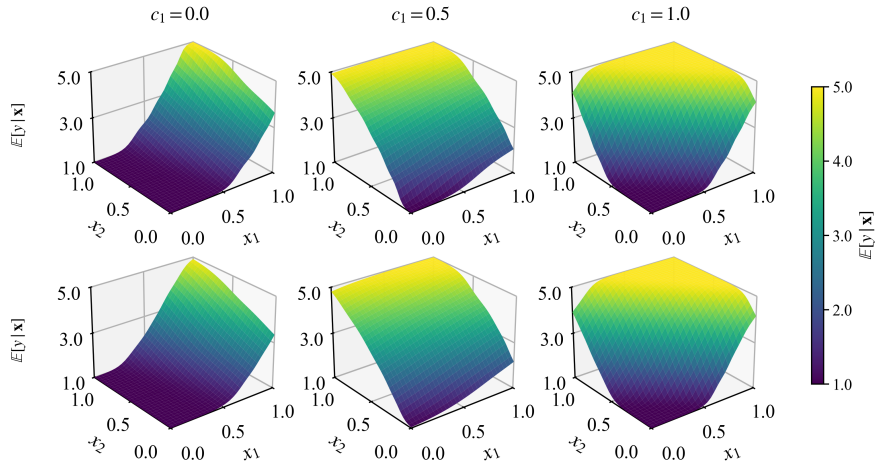

Figure 2: Coupled shape-regime recovery. Ground-truth (top) and five-seed mean CMRSN (bottom) expected ordinal responses at $c_1 = 0, 0.5, 1$ (columns), with $c_2 = 0.5, z = 0$.

Spatial recovery comparison. Fig. 3 localizes expected-label errors. CMRSN, MLP, CoMNN, and KFL remain accurate across contexts; Wc-MLP and A-PMNN show broad regime mismatch, whereas SMM and HLL have localized errors near the saturated diagonal transition. CMRSN's confined residuals and zero sampled MVR in Table 3 support multivariate geometric adaptation without relaxing partial monotonicity.

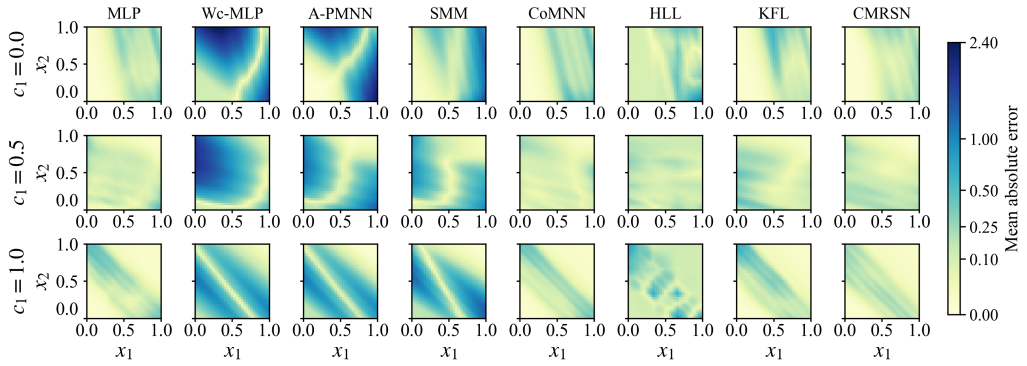

Figure 3: Five-seed mean absolute expected-label errors. Columns show methods; rows show $c_1 = 0, 0.5, 1$, with $c_2 = 0.5, z = 0$. All panels share a square-root color scale, with ticks in original error units.

5 Conclusion

CMRSN learns context-conditioned multivariate monotonic response surfaces through expert gates, smooth bases, and normalized nonnegative score directions. Restricting context modulation to these order-preserving components lets the response geometry vary across contexts while preserving exact partial monotonicity. The smooth and ordered Bernstein components provide complementary shape modeling, with the latter providing the constructive basis for universal approximation of continuous partially monotonic scores on compact domains. Experiments across regression, binary classification, and ordinal classification show competitive predictive–monotonicity trade-offs, while the controlled synthetic task demonstrates recovery of context-dependent surface geometry. Future work will explore sparse or factorized Bernstein representations to mitigate the tensor-product block's exponential scaling with the constrained dimension.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability and language of the manuscript, assist with experimental code implementation and debugging, and provide feedback on the design of the synthetic experiments. We did not use generative AI tools for other applicable tasks requiring disclosure; the remaining required-disclosure tasks are not applicable to this work. We reviewed all AI-assisted work, manually checked the manuscript, tested the assisted code, and independently reviewed the synthetic experimental design. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Code and data are available in an anonymous review repository at `https://anonymous.4open.science/r/CMRSN-ICLR2027-review-0CC7/`. The experimental pipeline includes the processed data splits, selected configurations, and scripts for the main comparison, ablations, and synthetic study. Preprocessing, metrics, split protocols, and synthetic generation follow Section 4 and Appendices B.1–B.3. Tuning uses seed 42, final experiments use seeds 43–47.

REFERENCES

Yaser S Abu-Mostafa. Learning from hints in neural networks. *Journal of Complexity*, 6(2):192–198, 1990.

Alon Arad and Saharon Rosset. Improving multi–class calibration through normalization–aware isotonic techniques. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 1574–1603, 2025.

Norman P. Archer and Shouhong Wang. Application of the back propagation neural network algorithm with monotonicity constraints for two–group classification problems. *Decision Sciences*, 24(1):60–75, 1993.

William Taylor Bakst, Nobuyuki Morioka, and Erez Louidor. Monotonic kronecker–factored lattice. In *International Conference on Learning Representations*, 2021.

Rudy R. Bunel, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, and Pawan K. Mudigonda. A unified view of piecewise linear neural network verification. In *Advances in Neural Information Processing Systems*, 2018.

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

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research*, 2009.

Matteo Francobaldi, Michele Lombardi, and Andrea Lodi. SMiLE: Provably enforcing global relational properties in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 37387–37395, 2026.

Akhil Gupta, Lavanya Marla, Ruoyu Sun, Naman Shukla, and Arinbjörn Kolbeinsson. PenDer: Incorporating shape constraints via penalized derivatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11536–11544, 2021.

Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander van Esbroeck. Monotonic calibrated interpolated look–up tables. *Journal of Machine Learning Research*, 17(109):1–47, 2016.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Christian Igel. Smooth min-max monotonic networks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20908–20923, 2024.

Seungwoo Jeong, Wonsik Jung, Junghyo Sohn, and Heung-Il Suk. Deep geometric learning with monotonicity constraints for alzheimer's disease progression. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):7090–7102, 2025.

Hyunho Kim and Jong-Seok Lee. Scalable monotonic neural networks. In *The Twelfth International Conference on Learning Representations*, pp. 35160–35176, 2024.

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

Mahdi Milani Fard, Kevin Canini, Andrew Cotter, Jan Pfeifer, and Maya Gupta. Fast and flexible monotonic functions with ensembles of lattices. In *Advances in Neural Information Processing Systems*, 2016.

João Monteiro, Mohamed Osama Ahmed, Hossein Hajimirsadeghi, and Greg Mori. Monotonicity regularization: Improved penalties and novel applications to disentangled representation learning and robust classification. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 1381–1391, 2022.

Niklas Nolte, Ouail Kitouni, and Mike Williams. Expressive monotonic neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Alejandro Polo-Molina, David Alfaya, and Jose Portela. A mathematical certification for positivity conditions in neural networks with applications to partial monotonicity and trustworthy AI. *IEEE Transactions on Neural Networks and Learning Systems*, 37(2):981–996, 2026a.

Alejandro Polo-Molina, David Alfaya, and Jose Portela. MonoKAN: Certified monotonic kolmogorov–arnold network. *Neural Networks*, 196:108278, 2026b.

Davor Runje and Sharath M Shankaranarayana. Constrained monotonic neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 29338–29353, 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*, pp. 52995–53018, 2025.

Joseph Sill. Monotonic networks. In *Advances in Neural Information Processing Systems*, 1997.

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

Joana Trindade, João Vinagre, Kelwin Fernandes, Nuno Paiva, and Alípio Jorge. Partially monotonic learning for neural networks. In *Advances in Intelligent Data Analysis XIX*, pp. 12–23, 2021.

Valentine Wargnier-Dauchelle, Thomas Grenier, Françoise Durand-Dubief, François Cotton, and Michaël Sdika. Explainable monotonic networks and constrained learning for interpretable classification and weakly supervised anomaly detection. *Pattern Recognition*, 160:111186, 2025.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Hiroki Yanagisawa, Kohei Miyaguchi, and Takayuki Katsuki. Hierarchical lattice layer for partially monotone neural networks. In *Advances in Neural Information Processing Systems*, pp. 11092–11103, 2022.

Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. In *Advances in Neural Information Processing Systems*, 2017.

Jiachi Zhao, Hongwen Zhang, Yue Wang, Yiteng Zhai, and Yao Yang. Deep isotonic embedding network: A flexible monotonic neural network. *Neural Networks*, 171:457–465, 2024.

A Theoretical Guarantees and Proofs

A.1 Exact partial monotonicity

Proposition 1 (Exact partial monotonicity)**.** *For any fixed $\mathbf{x}_U$, the score s_Θ in Eq.* (15) *is nondecreasing with respect to $\mathbf{x}_M$ on K_M. Therefore, CMRSN satisfies Eq.* (1)*. The regression prediction, binary positive-class probability and threshold prediction, and ordinal cumulative probabilities and threshold prediction are nondecreasing functions of this score.*

Proof. For fixed $\mathbf{x}_U$, the context $\mathbf{c} = q_\varphi(\mathbf{x}_U)$ is constant with respect to $\mathbf{x}_M$. The basis functions in Eq. (6) are element-wise nondecreasing because their scales are strictly positive. Affine maps and residual paths with nonnegative weights preserve coordinate-wise monotonicity, so the smooth block is nondecreasing by induction over layers. On $[0,1]^{d_M}$, the partial derivative of a Bernstein polynomial is a nonnegative Bernstein basis expansion weighted by adjacent coefficient differences, which are nonnegative under Eq. (10). Composing with T_M preserves this order on K_M. The unified expert representation is therefore nondecreasing in $\mathbf{x}_M$, and Eq. (15) is a nonnegative weighted sum of such representations plus a context-only bias. Finally, regression uses a positive affine rescaling; the binary logistic link and its threshold decision, and the ordinal cumulative logistic links and ordered-threshold decision, are nondecreasing in the score. □

A.2 Monotonic Bernstein density

Lemma 1 (Density of monotonic Bernstein experts)**.** *Let $m : [0,1]^{d_M} \to \mathbb{R}$ be continuous and nondecreasing in every coordinate. For an integer $N \geq 1$, define the Bernstein polynomial*

$$B_N m(\mathbf{x}_M) = \sum_{\mathbf{k}\in\{0,\dots,N\}^{d_M}} m(\mathbf{k}/N) \prod_{j=1}^{d_M} \binom{N}{k_j} x_{M,j}^{k_j} (1-x_{M,j})^{N-k_j}. \tag{23}$$

For any $\varepsilon > 0$, there exists an integer N such that $\sup_{\mathbf{x}_M\in[0,1]^{d_M}} |B_N m(\mathbf{x}_M) - m(\mathbf{x}_M)| < \varepsilon$. Moreover, $B_N m$ is nondecreasing in every coordinate.

Proof. For $k = 0,\dots,N$, let $b_{k,N}(x) = \binom{N}{k} x^k (1-x)^{N-k}$ denote the univariate Bernstein basis. For a multi-index $\mathbf{k} = (k_1,\dots,k_{d_M})$, write $b_{\mathbf{k},N}(\mathbf{x}_M) = \prod_{j=1}^{d_M} b_{k_j,N}(x_{M,j})$, so the Bernstein polynomial in Eq. (23) is $B_N m(\mathbf{x}_M) = \sum_{\mathbf{k}} m(\mathbf{k}/N) b_{\mathbf{k},N}(\mathbf{x}_M)$.

First, $B_N m$ converges uniformly to m. Let $Z = (Z_1,\dots,Z_{d_M})$, where $Z_1,\dots,Z_{d_M}$ are independent binomial random variables with $Z_j \sim \mathrm{Binomial}(N, x_{M,j})$. Then

$$B_N m(\mathbf{x}_M) = \mathbb{E}[m(Z_1/N,\dots,Z_{d_M}/N)]\,.$$

Because m is continuous on the compact cube, it is uniformly continuous. For any $\eta > 0$, choose $\delta > 0$ such that $\|\mathbf{z}-\mathbf{x}_M\|_\infty < \delta$ implies $|m(\mathbf{z})-m(\mathbf{x}_M)| < \eta$. Let $M_m = \sup_{\mathbf{x}_M\in[0,1]^{d_M}} |m(\mathbf{x}_M)|$. The union bound and Chebyshev's inequality give

$$\mathbb{P}(\|Z/N - \mathbf{x}_M\|_\infty \geq \delta) \leq \sum_{j=1}^{d_M} \frac{x_{M,j}(1-x_{M,j})}{N\delta^2} \leq \frac{d_M}{4N\delta^2}.$$

Thus $|B_N m(\mathbf{x}_M) - m(\mathbf{x}_M)| \leq \eta + 2M_m \frac{d_M}{4N\delta^2}$, uniformly in $\mathbf{x}_M$. Taking N large proves uniform convergence.

Second, $B_N m$ is coordinate-wise nondecreasing. For coordinate j,

$$\frac{\partial B_N m(\mathbf{x}_M)}{\partial x_{M,j}} = N \sum_{\mathbf{k}:k_j\leq N-1} [m((\mathbf{k}+\mathbf{e}_j)/N) - m(\mathbf{k}/N)]\, b_{k_j,N-1}(x_{M,j}) \prod_{\nu\neq j} b_{k_\nu,N}(x_{M,\nu}),$$

where the sum ranges over $k_j = 0,\dots,N-1$ and $k_\nu = 0,\dots,N$ for $\nu \neq j$, and $\mathbf{e}_j$ is the jth unit vector. Each adjacent coefficient difference is nonnegative because m is nondecreasing in coordinate j, and each basis term is nonnegative. Therefore every partial derivative is nonnegative, so $B_N m$ is nondecreasing in every coordinate. □

A.3 UNIVERSAL APPROXIMATION

Theorem 1 (Universal approximation of partially monotonic scores)**.** *Let $K_M \subset \mathbb{R}^{d_M}$ be a nondegenerate compact rectangle and let $K_U \subset \mathbb{R}^{d_U}$ be compact. Let $f^\star : K_M \times K_U \to \mathbb{R}$ be continuous and nondecreasing in $\mathbf{x}_M$ for every fixed $\mathbf{x}_U$. Assume that, with sufficiently large context dimension, the context encoder and linear gate heads can uniformly approximate continuous vector-valued logit maps on K_U. Then, for every $\varepsilon > 0$, there exists a finite CMRSN using the ordered Bernstein block in Eq.* (9) *such that*

$$\sup_{(\mathbf{x}_M,\mathbf{x}_U)\in K_M\times K_U} |s_\Theta(\mathbf{x}_M,\mathbf{x}_U) - f^\star(\mathbf{x}_M,\mathbf{x}_U)| < \varepsilon. \tag{24}$$

Proof. By an affine coordinate-wise rescaling, assume without loss of generality that $K_M = [0,1]^{d_M}$. Fix $\varepsilon > 0$. Since $f^\star$ is continuous on the compact set $K_M \times K_U$, it is uniformly continuous. Hence there exists a finite open cover $\{\mathcal{U}_e\}_{e=1}^E$ of K_U and representatives $\mathbf{u}_e \in \mathcal{U}_e$ such that

$$\sup_{\mathbf{x}_M\in K_M} |f^\star(\mathbf{x}_M,\mathbf{x}_U) - f^\star(\mathbf{x}_M,\mathbf{u}_e)| < \varepsilon/4 \tag{25}$$

whenever $\mathbf{x}_U \in \mathcal{U}_e$. Let $\{\alpha_e\}_{e=1}^E$ be a continuous partition of unity subordinate to this cover, so $\alpha_e(\mathbf{x}_U) \geq 0$ and $\sum_{e=1}^E \alpha_e(\mathbf{x}_U) = 1$ for every $\mathbf{x}_U \in K_U$, with $\operatorname{supp}(\alpha_e) \subseteq \mathcal{U}_e$.

For each representative $\mathbf{u}_e$, the slice $f^\star(\cdot,\mathbf{u}_e)$ is continuous and nondecreasing on K_M. Because the cover is finite and the Bernstein sequence converges uniformly for every slice, choose a single sufficiently large degree N such that, for every e, the monotonic Bernstein expert $m_e(\mathbf{x}_M) = B_N[f^\star(\cdot,\mathbf{u}_e)](\mathbf{x}_M)$ satisfies

$$\sup_{\mathbf{x}_M\in K_M} |m_e(\mathbf{x}_M) - f^\star(\mathbf{x}_M,\mathbf{u}_e)| < \varepsilon/4. \tag{26}$$

Let $B_m = \max_{e=1,\dots,E} \sup_{\mathbf{x}_M\in K_M} |m_e(\mathbf{x}_M)| < \infty$ and choose $\delta = \varepsilon/(4E(B_m+1))$.

It remains to approximate the partition of unity by softmax gates. For any small $\gamma > 0$, the smoothed partition weights are defined as

$$\tilde{\alpha}_e(\mathbf{x}_U) = \frac{\alpha_e(\mathbf{x}_U) + \gamma}{1 + E\gamma}.$$

Then $\tilde{\alpha}_e > 0$ and $\sup_{\mathbf{x}_U} |\tilde{\alpha}_e(\mathbf{x}_U) - \alpha_e(\mathbf{x}_U)|$ can be made arbitrarily small by choosing γ small. Since $\log \tilde{\alpha}_e(\mathbf{x}_U)$ is continuous, the assumed context-gate logit class can approximate the vector-valued logit map $\mathbf{x}_U \mapsto (\log \tilde{\alpha}_1(\mathbf{x}_U), \dots, \log \tilde{\alpha}_E(\mathbf{x}_U))$ uniformly. The softmax map is continuous, so the resulting gates can be made uniformly close to $\tilde{\alpha}_e$. Together with the bound between $\tilde{\alpha}_e$ and α_e, the triangle inequality implies that choosing γ and the logit approximation error sufficiently small yields gates satisfying

$$\sup_{\mathbf{x}_U\in K_U} |\pi_e(q_\varphi(\mathbf{x}_U)) - \alpha_e(\mathbf{x}_U)| < \delta \quad \text{for all } e. \tag{27}$$

Using these gates and the slice-wise Bernstein experts, the Bernstein-coordinate score is defined as

$$\widetilde{s}(\mathbf{x}_M,\mathbf{x}_U) = \sum_{e=1}^E \pi_e(q_\varphi(\mathbf{x}_U)) m_e(\mathbf{x}_M). \tag{28}$$

This score can be realized as the Bernstein-coordinate contribution under the normalized readout. Fix any continuous smooth-coordinate maps, which are bounded on the compact domain, and let $B_S = \max_e \sup_{\mathbf{x}_M,\mathbf{x}_U} \|\mathbf{z}_e^S(\mathbf{x}_M; q_\varphi(\mathbf{x}_U))\|_\infty < \infty$. Write $v_{e,B} := v_{e,n_L+1}$ for the Bernstein-coordinate weight. Choose $0 < \rho < 1$ with $\rho B_S < \varepsilon/4$. Set the constant direction weights to $v_{e,B} = 1-\rho$ and $v_{e,j} = \rho/n_L$ for $j = 1,\dots,n_L$. These strictly positive weights sum to one and are realized exactly by constant softmax logits equal to their logarithms. Scale the Bernstein coefficients of expert e by $1/(1-\rho)$; this positive scaling preserves their order and makes its weighted Bernstein coordinate exactly m_e. Set the context bias to zero. The remaining smooth contribution has absolute

value at most $\rho B_S < \varepsilon/4$. It remains to bound the displayed Bernstein-coordinate score:

$$\begin{aligned}|\widetilde{s}(\mathbf{x}_M, \mathbf{x}_U) - f^\star(\mathbf{x}_M, \mathbf{x}_U)| &\le \sum_{e=1}^{E} \alpha_e(\mathbf{x}_U)|m_e(\mathbf{x}_M) - f^\star(\mathbf{x}_M, \mathbf{u}_e)| \\ &\quad + \sum_{e=1}^{E} \alpha_e(\mathbf{x}_U)|f^\star(\mathbf{x}_M, \mathbf{u}_e) - f^\star(\mathbf{x}_M, \mathbf{x}_U)| \\ &\quad + \sum_{e=1}^{E} |\pi_e(q_\varphi(\mathbf{x}_U)) - \alpha_e(\mathbf{x}_U)|\,|m_e(\mathbf{x}_M)| \\ &< \varepsilon/4 + \varepsilon/4 + \varepsilon/4 = 3\varepsilon/4. \end{aligned} \tag{29}$$

After adding the smooth-coordinate contribution, the full CMRSN score is within ε of $f^\star$. The second term is bounded because $\alpha_e(\mathbf{x}_U)$ is nonzero only inside $\mathcal{U}_e$. Since the gates are nonnegative constants with respect to $\mathbf{x}_M$ for fixed $\mathbf{x}_U$, and each m_e is nondecreasing in $\mathbf{x}_M$, the constructed approximation is partially monotonic. □

B Experimental Details

B.1 Datasets and preprocessing

Datasets and feature partitions. Table 4 lists the eight benchmarks and their imposed monotonic directions. Regression retains the original continuous targets, whereas ordinal classification uses native ordered classes reindexed as $1, \dots, C$. For binary classification, the positive class denotes two-year recidivism in COMPAS and annual income exceeding $50,000 in Adult.

Table 4: Benchmarks and domain-specified feature partitions. n is the dataset size, d the processed input dimension, d_M the constrained dimension, and C the number of classes. Signs refer to the supplied feature encoding before sign alignment. All remaining features constitute $\mathbf{x}_U$.

Dataset	n	d	d_M	C	Constrained features and directions
BlogFeedback	60,021	276	8	–	Attributes 051–054 and 056–059 $(+)$
Concrete Strength	1,030	8	3	–	Cement $(+)$, water $(-)$, age $(+)$
Auto MPG	392	9	3	–	Displacement $(-)$, horsepower $(-)$, weight $(-)$
COMPAS	6,172	13	4	2	Priors count, juvenile felony, misdemeanor, and other counts $(+)$
Adult	48,842	108	2	2	Education number $(+)$, capital gain $(+)$
Wine Quality	1,359	11	6	6	Volatile acidity $(-)$, citric acid $(+)$, chlorides $(-)$, density $(-)$, sulphates $(+)$, alcohol $(+)$
Car Evaluation	1,728	6	6	4	Buying, maintenance, doors, persons, luggage boot, safety: all encoded coordinates $(+)$
Maternal Health Risk	1,014	6	4	3	Age, systolic and diastolic blood pressure, blood sugar $(+)$

Preprocessing. Numerical inputs use training-only min–max scaling and clipping to $[0, 1]$, followed by sign alignment. Thus, $K_M = [0, 1]^{d_M}$ and T_M is the identity on the model-input domain. Clipping preserves coordinate-wise order but makes predictions constant along a constrained coordinate beyond its training bounds.

B.2 Comparison and training details

Comparison methods. Table 5 summarizes the implemented comparison. All methods output a scalar score and use the task-specific prediction rules from Section 3.3.

Model selection. Each real-data dataset–split–method combination evaluates 12 candidate configurations under a common search budget. Baseline capacities approximately match the corresponding CMRSN backbone parameter counts within their available architecture pools. For each candidate, the checkpoint with the lowest validation loss is retained. Configuration selection minimizes validation NRMSE for regression, maximizes validation AUROC for binary classification, and minimizes validation NMAE for ordinal classification.

Table 5: Comparison methods and their implementations.

Method	Implementation
MLP (Hornik et al., 1989)	Unrestricted neural predictor without structural monotonicity constraints.
Wc-MLP	Nonnegative-weight ReLU MLP, monotonic in all supplied input coordinates.
A-PMNN (Daniels & Velikova, 2010; Trindade et al., 2021)	Explicit additive implementation combining a monotonic branch over $\mathbf{x}_M$ with an unrestricted branch over $\mathbf{x}_U$.
SMM (Igel, 2024)	Smooth min–max network with sign-constrained weights on $\mathbf{x}_M$ and an auxiliary network on $\mathbf{x}_U$.
CoMNN (Runje & Shankaranarayana, 2023)	Constrained monotonic network using specialized activations and feature-wise monotonicity indicators.
HLL (Yanagisawa et al., 2022)	Hierarchical lattice layer with a context network supplying vertex parameters when context is present.
KFL (Bakst et al., 2021)	Kronecker-factored lattice with structural monotonicity constraints on selected inputs.
CMRSN	Each expert combines context-modulated smooth bases with an ordered Bernstein block; context-dependent softmax weights aggregate within and across experts.

Monotonicity evaluation. MVR pairs are constructed by sampling preprocessed evaluation inputs and replacing one constrained coordinate x_j with $x_j + u(1 - x_j)$, where $u \sim \mathrm{Uniform}[0, 1]$, while holding all other coordinates, including context, fixed. Perturbed coordinates are balanced across pairs.

Optimization and implementation. Real-data models train for at most 300 epochs, with early-stopping patience 60 and minimum validation-loss improvement 10^{-6}. These settings equalize the epoch cap, but not gradient-update counts or computation; the comparison therefore reflects both architecture and method-specific optimization settings. PyTorch methods use Adam with coupled weight decay, whereas KFL uses TensorFlow AdamW. CMRSN uses $L = 1$, $J_1 = 1$, and $m_1 = n_1 = d_h$, with a two-hidden-layer ReLU context encoder of width d_c. The implementation adds 10^{-6} to softplus-constrained smooth weights and basis scales. After each Adam update, Bernstein repair is applied outside automatic differentiation, while Adam's first- and second-moment states are retained. The reported runs use two complete axis sweeps according to Eq. (12); one sweep already computes the multidimensional prefix maximum, so the second leaves the repaired coefficients unchanged. The repair enforces coefficient order but is not a Euclidean projection.

B.3 Synthetic task and evaluation

Synthetic mechanism task. Let x_1, x_2, c_1, c_2 be independent uniform variables on $[0, 1]$, and let z be an independent Bernoulli variable with probability $1/2$. The constrained input is $\mathbf{x}_M = (x_1, x_2)$ and the context is $\mathbf{x}_U = (c_1, c_2, z)$. Define the positive projections $u = 0.8x_1 + 0.2x_2$, $v = 0.2x_1 + 0.8x_2$, and $w = 0.5(x_1 + x_2)$.

With $a = 2 + 4c_2$, $b = 8 + 4c_2$, and $t = 0.4 + 0.2z$, the three score surfaces are

$$m_{\mathrm{cvx}} = u^2, \qquad m_{\mathrm{ccv}} = \frac{\log(1 + av)}{\log(1 + a)}, \\ m_{\mathrm{sat}} = \frac{\mathrm{sigmoid}(b(w - t)) - \mathrm{sigmoid}(-bt)}{\mathrm{sigmoid}(b(1 - t)) - \mathrm{sigmoid}(-bt)}. \tag{30}$$

Let $h(q) = q^2(3 - 2q)$ for $q \in [0, 1]$ and $\mathrm{clip}(q) = \min(1, \max(0, q))$. Set $L(c_1) = h(\mathrm{clip}(2c_1))$ and $R(c_1) = h(\mathrm{clip}(2c_1 - 1))$. The ground-truth score is

$$s^\star(\mathbf{x}) = (1 - L(c_1))m_{\mathrm{cvx}} + (L(c_1) - R(c_1))m_{\mathrm{ccv}} + R(c_1)m_{\mathrm{sat}}. \tag{31}$$

These mixing weights are nonnegative and sum to one. They isolate the convex, concave, and S-shaped saturated score regimes at $c_1 = 0, 0.5, 1$, respectively, and interpolate smoothly between them. Each projection and each shape function is nondecreasing in both constrained coordinates. The resulting score is therefore partially monotonic, with context changing both its shape and its relative response to x_1 and x_2.

Five-class labels are sampled using

$$P^\star(y > k \mid \mathbf{x}) = \mathrm{sigmoid}\left(\frac{s^\star(\mathbf{x}) - \vartheta_k^\star}{T_{\mathrm{syn}}}\right), \qquad k = 1, \ldots, 4. \tag{32}$$

The thresholds are the empirical 20th, 40th, 60th, and 80th percentiles of $200{,}000$ calibration scores generated with seed 2026. The temperature is $T_{\mathrm{syn}} = \max(0.15\,\mathrm{std}(s^\star), 10^{-3})$ on the same calibration sample. The calibration sample is disjoint from all training, validation, and test sets and is held fixed across methods and seeds.

Tuning uses independently generated training and validation sets with seed 42. For each final seed in $\{43, 44, 45, 46, 47\}$, independent training, validation, and test sets contain $2{,}000$, $1{,}000$, and $5{,}000$ samples, respectively; all methods share the sets for that seed. No outer-fold cross-validation is used. Each method evaluates 12 candidate configurations. All synthetic runs use batch size 64, at most 300 epochs, early-stopping patience 60, and $10{,}000$ pairs per MVR estimate. Checkpoints are chosen by validation ordinal logistic loss; hyperparameter configurations are selected by validation CP-RMSE, breaking ties by validation NMAE and loss.

Synthetic recovery metrics. For a test set of size n_{te}, let $p^\star_{ik}$ and $\hat{p}_{ik}$ denote the true and predicted cumulative probabilities. We compute

$$\mathrm{CP\text{-}RMSE} = \sqrt{\frac{1}{n_{\mathrm{te}}(C-1)} \sum_{i=1}^{n_{\mathrm{te}}} \sum_{k=1}^{C-1} (\hat{p}_{ik} - p^\star_{ik})^2}. \tag{33}$$

For visualization, each context $(c_1, c_2, z) \in \{(0, 0.5, 0), (0.5, 0.5, 0), (1, 0.5, 0)\}$ uses a 31×31 grid over (x_1, x_2). Surf. MAE is the mean absolute error between $\mathbb{E}_{\widehat{\Theta}}[y \mid \mathbf{x}]$ and $\mathbb{E}^\star[y \mid \mathbf{x}]$ over these $3 \cdot 31^2$ grid points, computed separately for each seed. The reported value is its five-seed mean and sample standard deviation. The figures visualize expected ordinal responses after the ordinal link; the convex, concave, and saturated regimes refer to the generating score functions in Eq. (30).

C Complete Experimental Results

C.1 Real-data comparison

Tables 6–8 report the per-dataset results underlying Table 1. Values are mean $\pm$ standard deviation across five seeds after averaging the five splits within each seed. For regression, the additional MAE values retain each dataset's original target units.

Table 6: Regression comparison by dataset.

BlogFeedback

Method	NRMSE ↓	MAE ↓	MVR ↓
MLP	0.7247 ± 0.0124	**6.4895 ± 0.2018**	0.2074 ± 0.0184
Wc-MLP	0.7165 ± 0.0007	7.4766 ± 0.1561	0.0000
A-PMNN	0.7122 ± 0.0141	6.9301 ± 0.0789	0.0000
SMM	0.7146 ± 0.0079	7.4865 ± 0.1108	0.0000
CoMNN	0.7002 ± 0.0136	6.5292 ± 0.2429	0.0000
HLL	0.6970 ± 0.0044	6.9522 ± 0.1050	0.0000
KFL	0.9037 ± 0.1852	7.5413 ± 0.3625	0.0000
CMRSN	**0.6960 ± 0.0164**	6.6627 ± 0.3954	0.0000

Concrete Strength

Method	NRMSE ↓	MAE ↓	MVR ↓
MLP	**0.3126 ± 0.0046**	**3.7185 ± 0.0771**	0.0737 ± 0.0105
Wc-MLP	0.6301 ± 0.0004	8.4090 ± 0.0122	0.0000
A-PMNN	0.6012 ± 0.0026	7.9531 ± 0.0469	0.0000
SMM	0.3996 ± 0.0020	4.9750 ± 0.0206	0.0000
CoMNN	0.4429 ± 0.0423	5.8038 ± 0.5921	0.0000
HLL	0.7032 ± 0.0022	9.4686 ± 0.0271	0.0000
KFL	0.4730 ± 0.0008	6.2230 ± 0.0077	0.0000
CMRSN	0.3891 ± 0.0080	4.7833 ± 0.1254	0.0000

Auto MPG

Method	NRMSE ↓	MAE ↓	MVR ↓
MLP	**0.3611 ± 0.0046**	**1.9666 ± 0.0314**	0.1866 ± 0.0420
Wc-MLP	0.4000 ± 0.0011	2.3355 ± 0.0133	0.0000
A-PMNN	0.3866 ± 0.0045	2.2078 ± 0.0362	0.0000
SMM	0.4146 ± 0.0048	2.2944 ± 0.0232	0.0000
CoMNN	0.4226 ± 0.0044	2.3598 ± 0.0258	0.0000
HLL	0.6266 ± 0.0054	3.7455 ± 0.0435	0.0000
KFL	0.3718 ± 0.0015	2.0551 ± 0.0067	0.0000
CMRSN	0.4013 ± 0.0118	2.2137 ± 0.0636	0.0000

Table 7: Binary comparison by dataset.

COMPAS

Method	AUROC ↑	Acc. ↑	MVR ↓
MLP	0.7351 ± 0.0002	0.6823 ± 0.0027	0.1832 ± 0.0199
Wc-MLP	0.6859 ± 0.0002	0.6504 ± 0.0001	0.0000
A-PMNN	0.7280 ± 0.0010	0.6756 ± 0.0011	0.0000
SMM	0.7344 ± 0.0005	0.6814 ± 0.0015	0.0000
CoMNN	**0.7363 ± 0.0008**	**0.6838 ± 0.0011**	0.0000
HLL	0.7119 ± 0.0031	0.6265 ± 0.0076	0.0000
KFL	0.7309 ± 0.0002	0.6811 ± 0.0013	0.0000
CMRSN	0.7327 ± 0.0005	0.6804 ± 0.0013	0.0000

Adult

Method	AUROC ↑	Acc. ↑	MVR ↓
MLP	0.9046 ± 0.0008	0.8516 ± 0.0008	0.0391 ± 0.0768
Wc-MLP	0.9053 ± 0.0001	0.8522 ± 0.0004	0.0000
A-PMNN	0.9050 ± 0.0003	0.8523 ± 0.0004	0.0000
SMM	0.9015 ± 0.0008	0.8480 ± 0.0007	0.0000
CoMNN	0.9073 ± 0.0004	0.8540 ± 0.0007	0.0000
HLL	0.8930 ± 0.0003	0.8399 ± 0.0004	0.0000
KFL	0.9070 ± 0.0003	**0.8557 ± 0.0003**	0.0000
CMRSN	**0.9076 ± 0.0010**	0.8549 ± 0.0007	0.0000

Table 8: Ordinal comparison by dataset.

Wine Quality

Method	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
MLP	0.5765 ± 0.0056	0.0915 ± 0.0012	0.5277 ± 0.0114	0.3007 ± 0.0221
Wc-MLP	0.5648 ± 0.0006	0.0950 ± 0.0002	0.4741 ± 0.0037	0.0000
A-PMNN	0.5640 ± 0.0048	0.0955 ± 0.0009	0.4893 ± 0.0047	0.0000
SMM	0.5671 ± 0.0026	0.0945 ± 0.0005	0.4396 ± 0.0034	0.0000
CoMNN	**0.5918 ± 0.0039**	**0.0882 ± 0.0007**	0.5222 ± 0.0053	0.0000
HLL	0.5458 ± 0.0040	0.0995 ± 0.0008	0.3523 ± 0.0079	0.0000
KFL	0.5540 ± 0.0052	0.0982 ± 0.0012	**0.5438 ± 0.0049**	0.0000
CMRSN	0.5672 ± 0.0071	0.0945 ± 0.0013	0.4815 ± 0.0077	0.0000

Car Evaluation

Method	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
MLP	**0.9854 ± 0.0032**	**0.0049 ± 0.0011**	**0.9864 ± 0.0031**	0.1449 ± 0.0136
Wc-MLP	0.8128 ± 0.0015	0.0648 ± 0.0005	0.7928 ± 0.0031	0.0000
A-PMNN	0.8116 ± 0.0030	0.0650 ± 0.0012	0.7923 ± 0.0066	0.0000
SMM	0.9139 ± 0.0047	0.0289 ± 0.0016	0.9090 ± 0.0052	0.0000
CoMNN	0.9140 ± 0.0062	0.0302 ± 0.0021	0.8972 ± 0.0078	0.0000
HLL	0.7002 ± 0.0000	0.1383 ± 0.0000	0.0000 ± 0.0000	0.0000
KFL	0.8642 ± 0.0012	0.0456 ± 0.0004	0.8784 ± 0.0015	0.0000
CMRSN	0.8744 ± 0.0092	0.0446 ± 0.0030	0.8402 ± 0.0104	0.0000

Maternal Health Risk

Method	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
MLP	**0.6583 ± 0.0101**	**0.1863 ± 0.0056**	**0.6902 ± 0.0114**	0.2692 ± 0.0175
Wc-MLP	0.5627 ± 0.0099	0.2317 ± 0.0049	0.5923 ± 0.0078	0.0000
A-PMNN	0.5782 ± 0.0066	0.2232 ± 0.0034	0.6134 ± 0.0065	0.0000
SMM	0.5709 ± 0.0033	0.2254 ± 0.0017	0.5959 ± 0.0061	0.0000
CoMNN	0.5852 ± 0.0043	0.2156 ± 0.0021	0.6258 ± 0.0042	0.0000
HLL	0.5314 ± 0.0109	0.2460 ± 0.0034	0.5406 ± 0.0096	0.0000
KFL	0.6214 ± 0.0035	0.1921 ± 0.0016	0.6769 ± 0.0033	0.0000
CMRSN	0.6079 ± 0.0050	0.2068 ± 0.0025	0.6528 ± 0.0070	0.0000

C.2 Ablation results

Tables 9–11 provide the full results for the four reported variants. Values are mean ± standard deviation across five seeds after averaging the five splits within each seed.

Table 9: Regression ablation by dataset.

BlogFeedback

Variant	NRMSE ↓	MAE ↓	MVR ↓
Full CMRSN	0.6960 ± 0.0164	6.6627 ± 0.3954	0.0000
w/o Surface Conditioning	0.6954 ± 0.0120	7.0882 ± 0.1633	0.0000
w/o Basis Modulation	**0.6855 ± 0.0070**	6.3906 ± 0.2497	0.0000
w/o Bernstein Block	0.7068 ± 0.0111	6.6821 ± 0.1903	0.0000
w/o Smooth Block	0.6878 ± 0.0047	**6.2400 ± 0.2201**	0.0000

Concrete Strength

Variant	NRMSE ↓	MAE ↓	MVR ↓
Full CMRSN	0.3891 ± 0.0080	4.7833 ± 0.1254	0.0000
w/o Surface Conditioning	0.5362 ± 0.0027	6.9274 ± 0.0325	0.0000
w/o Basis Modulation	0.4713 ± 0.0127	6.0434 ± 0.1910	0.0000
w/o Bernstein Block	**0.3819 ± 0.0081**	**4.6609 ± 0.1195**	0.0000
w/o Smooth Block	0.4798 ± 0.0048	6.1378 ± 0.0816	0.0000

Auto MPG

Variant	NRMSE ↓	MAE ↓	MVR ↓
Full CMRSN	0.4013 ± 0.0118	2.2137 ± 0.0636	0.0000
w/o Surface Conditioning	0.4235 ± 0.0066	2.3465 ± 0.0451	0.0000
w/o Basis Modulation	0.4111 ± 0.0054	2.2644 ± 0.0357	0.0000
w/o Bernstein Block	0.3964 ± 0.0134	2.1762 ± 0.0763	0.0000
w/o Smooth Block	**0.3837 ± 0.0047**	**2.0802 ± 0.0387**	0.0000

Table 10: Binary ablation by dataset.

COMPAS

Variant	AUROC ↑	Acc. ↑	MVR ↓
Full CMRSN	0.7327 ± 0.0005	0.6804 ± 0.0013	0.0000
w/o Surface Conditioning	0.7306 ± 0.0004	0.6752 ± 0.0018	0.0000
w/o Basis Modulation	0.7324 ± 0.0006	0.6786 ± 0.0023	0.0000
w/o Bernstein Block	**0.7345 ± 0.0006**	**0.6810 ± 0.0018**	0.0000
w/o Smooth Block	0.7334 ± 0.0005	0.6791 ± 0.0022	0.0000

Adult

Variant	AUROC ↑	Acc. ↑	MVR ↓
Full CMRSN	**0.9076 ± 0.0010**	**0.8549 ± 0.0007**	0.0000
w/o Surface Conditioning	0.9073 ± 0.0005	0.8541 ± 0.0006	0.0000
w/o Basis Modulation	0.9070 ± 0.0009	0.8543 ± 0.0007	0.0000
w/o Bernstein Block	0.9074 ± 0.0009	0.8545 ± 0.0007	0.0000
w/o Smooth Block	0.9051 ± 0.0013	0.8527 ± 0.0011	0.0000

Table 11: Ordinal ablation by dataset.

Wine Quality

Variant	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
Full CMRSN	**0.5672 ± 0.0071**	**0.0945 ± 0.0013**	**0.4815 ± 0.0077**	0.0000
w/o Surface Conditioning	0.5656 ± 0.0055	0.0948 ± 0.0010	0.4784 ± 0.0060	0.0000
w/o Basis Modulation	0.5651 ± 0.0062	0.0954 ± 0.0015	0.4722 ± 0.0135	0.0000
w/o Bernstein Block	0.5654 ± 0.0053	0.0950 ± 0.0010	0.4791 ± 0.0080	0.0000
w/o Smooth Block	0.5503 ± 0.0036	0.0993 ± 0.0008	0.3893 ± 0.0080	0.0000

Car Evaluation

Variant	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
Full CMRSN	0.8744 ± 0.0092	0.0446 ± 0.0030	**0.8402 ± 0.0104**	0.0000
w/o Surface Conditioning	0.8194 ± 0.0088	0.0631 ± 0.0030	0.7779 ± 0.0120	0.0000
w/o Basis Modulation	0.8434 ± 0.0057	0.0552 ± 0.0018	0.7979 ± 0.0064	0.0000
w/o Bernstein Block	0.8628 ± 0.0134	0.0482 ± 0.0047	0.8351 ± 0.0193	0.0000
w/o Smooth Block	**0.8827 ± 0.0009**	**0.0426 ± 0.0003**	0.8344 ± 0.0007	0.0000

Maternal Health Risk

Variant	Acc. ↑	NMAE ↓	QWK ↑	MVR ↓
Full CMRSN	0.6079 ± 0.0050	**0.2068 ± 0.0025**	**0.6528 ± 0.0070**	0.0000
w/o Surface Conditioning	0.5766 ± 0.0128	0.2221 ± 0.0072	0.6110 ± 0.0107	0.0000
w/o Basis Modulation	0.5771 ± 0.0077	0.2214 ± 0.0039	0.6094 ± 0.0052	0.0000
w/o Bernstein Block	**0.6081 ± 0.0122**	0.2074 ± 0.0063	0.6518 ± 0.0110	0.0000
w/o Smooth Block	0.5960 ± 0.0037	0.2130 ± 0.0015	0.6193 ± 0.0028	0.0000

C.3 Selected model sizes

Table 12 reports the actual selected parameter counts. Under the capacity-targeted search, CMRSN's split-averaged parameter count is 0.39–1.20 times the median of the seven baselines on the same dataset, so it does not have a systematic size advantage. Individual configurations are not exactly parameter matched, notably HLL on Car Evaluation, and parameter counts do not measure computational cost. Taken together, these results indicate that CMRSN's competitive performance is not simply explained by using substantially more parameters.

Table 12: Trainable parameters in the selected models, including prediction heads. Ranges span the five data splits; a single number denotes an identical count across splits.

Method	BlogFeedback	Concrete Strength	Auto MPG	COMPAS
MLP	28,721–59,873	12,977–26,561	2,081–26,753	3,009–27,521
Wc-MLP	28,721	2,689–26,561	2,081–26,753	3,009
A-PMNN	26,843–56,675	2,499–26,659	13,699–26,771	2,659–27,219
SMM	30,192–53,902	29,442	2,410–11,266	2,578–26,306
CoMNN	43,777–59,873	2,689	2,081–13,121	3,009–27,521
HLL	33,858	12,509–25,538	12,605–25,666	25,794
KFL	26,529–53,057	12,801–25,601	2,369–28,673	13,569–27,137
CMRSN	29,869–59,565	12,101–26,157	2,365–26,181	12,509–26,661
Method	Adult	Wine Quality	Car Evaluation	Maternal Health Risk
MLP	2,621	4,998–27,846	5,380–16,676	2,563–26,179
Wc-MLP	2,621–12,657	4,998–14,118	5,380–30,500	2,563
A-PMNN	2,339	5,432–29,864	4,677	2,437–26,437
SMM	2,602–13,034	5,367–29,799	16,133–28,677	12,868–29,700
CoMNN	27,377	4,998–27,846	5,380–30,500	2,563–26,179
HLL	14,214–28,422	4,551–25,543	69	2,773–27,349
KFL	2,601–13,857	5,766–34,822	4,868	2,435–12,803
CMRSN	2,629	5,366–29,546	5,348–29,448	2,631–12,399