Neural Additive Experts: Context-Gated Experts for Controllable Model Additivity

Guangzhi Xiong
University of Virginia

Sanchit Sinha
University of Virginia

Aidong Zhang
University of Virginia

Abstract

The trade-off between interpretability and accuracy remains a core challenge in machine learning. Standard Generalized Additive Models (GAMs) offer clear feature attributions but are often constrained by their strictly additive nature, which can limit predictive performance. Introducing feature interactions can boost accuracy yet may obscure individual feature contributions. To address these issues, we propose Neural Additive Experts (NAEs), a novel framework that seamlessly balances interpretability and accuracy. NAEs employ a mixture of experts framework, learning multiple specialized networks per feature, while a dynamic gating mechanism integrates information across features, thereby relaxing rigid additive constraints. Furthermore, we propose targeted regularization techniques to mitigate variance among expert predictions, facilitating a smooth transition from an exclusively additive model to one that captures intricate feature interactions while maintaining clarity in feature attributions. Our theoretical analysis and experiments on synthetic data illustrate the model's flexibility, and extensive evaluations on real-world datasets confirm that NAEs achieve an optimal balance between predictive accuracy and transparent, feature-level explanations. The code is available at `https://github.com/Teddy-XiongGZ/NAE`.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

The tension between interpretability and predictive accuracy is a central challenge in modern machine learning. While deep neural networks and other complex models have achieved remarkable success across a range of domains, their opaque decision-making processes often hinder adoption in high-stakes settings where transparency and trust are paramount. In contrast, Generalized Additive Models (GAMs) (Hastie, 2017; Agarwal et al., 2021; Lou et al., 2012; Chang et al., 2022) are widely valued for their interpretability, the degree to which an observer can understand the cause of a decision (Biran and Cotton, 2017; Miller, 2019), as they decompose predictions into clear, feature-specific contributions (McIntosh, 2025; Ibrahim et al., 2025; Zhang et al., 2024; Bouchiat et al., 2023). However, the strictly additive structure of GAMs can limit their ability to capture complex relationships among features, often resulting in suboptimal predictive performance.

A common approach to improve the predictive accuracy of GAMs is to incorporate feature interactions (Ruppert, 2004; Lou et al., 2013). However, introducing such interactions often diminishes interpretability, as it becomes more difficult to transparently attribute predictions to individual features, since interpreting a single feature may require a comprehensive analysis of multiple related feature interactions (Lou et al., 2012; Caruana et al., 2015; Christoph, 2020). Furthermore, GAMs with interactions typically lack explicit mechanisms to control model additivity, the degree to which outputs can be decomposed into additive feature contributions, which is a hallmark of standard GAMs. This limitation restricts users' ability to flexibly balance accuracy and interpretability to suit specific application requirements.

In this work, we propose *Neural Additive Experts* (NAEs), a novel framework designed to address this fundamental trade-off. NAEs extend the classical additive paradigm by associating each feature with a set of specialized expert networks. A dynamic gating

mechanism adaptively integrates information across features, allowing the model to relax the rigid additive constraints when interactions are important. To control the balance between accuracy and interpretability, we introduce a targeted regularization term that controls the variance among expert predictions, enabling a smooth transition from a purely additive model to one that captures nuanced feature interactions while maintaining clarity in feature attributions.

Our theoretical analysis demonstrates how NAEs can recover complex data-generating processes that are inaccessible to standard GAMs, and provides insight into the role of regularization in balancing model flexibility and interpretability. Through experiments on synthetic and real-world datasets, we show that NAEs achieve competitive accuracy with state-of-the-art black-box models, while offering transparent, feature-level explanations. By providing a principled mechanism to control the trade-off between interpretability and accuracy, NAEs offer a practical solution for applications where both predictive performance and transparency are essential.

The contributions of this work are summarized as follows:

- We introduce **Neural Additive Experts** (NAEs), a novel framework that extends additive models with a mixture-of-experts architecture and dynamic gating, enabling flexible integration of feature interactions while preserving interpretability.
- We propose targeted regularization techniques that allow users to control the variance among expert predictions, providing a principled mechanism to balance model flexibility and transparency.
- We provide theoretical analysis demonstrating how NAEs can recover complex data-generating processes that are inaccessible to standard GAMs, and clarify the role of regularization in managing the trade-off between accuracy and interpretability.
- We empirically validate NAEs on both synthetic and real-world datasets, showing that our approach achieves competitive predictive performance with state-of-the-art black-box models while maintaining clear, feature-level explanations.

2 NEURAL ADDITIVE EXPERTS

Neural Additive Expert (NAE) is a framework that extends classical GAMs by associating each feature with a set of expert predictors and employing a dynamic, context-aware gating mechanism. This architecture enables the model to capture complex, context-dependent feature effects while preserving the additive structure that supports interpretability. Figure 1 provides an overview of the NAE architecture.

2.1 Feature Representation and Expert Networks

Let x_i $(i = 1, \ldots, n)$ denote the i-th feature of an input instance. Each feature is first mapped into a latent representation via an encoder:

$$\mathcal{E}_i : \mathcal{X}_i \to \mathbb{R}^d, \tag{1}$$

where $\mathcal{X}_i$ is the domain of x_i and d is the latent dimension. Unlike standard GAMs, which use a single predictor per feature, NAEs introduce a set of K expert networks for each feature. The k-th expert for feature x_i produces:

$$o_{ik} = g_{ik}\big(\mathcal{E}_i(x_i)\big), \tag{2}$$

where $g_{ik} : \mathbb{R}^d \to \mathbb{R}$ is typically a simple function (e.g., linear or shallow neural network) that maps the latent representation to a feature effect.

2.2 Dynamic Gating and Expert Aggregation

A central component of NAEs is the dynamic gating mechanism, which adaptively integrates information across features to determine the relevance of each expert. For each feature x_j, we compute a score vector to quantify the influence of the x_i value on the expert selection for feature x_j:

$$\varphi_j = \mu_j + \sum_{i=1}^{n} \mathcal{A}_{ij}^{\top} \mathcal{E}_i(x_i), \tag{3}$$

where $\mathcal{A}_{ij} \in \mathbb{R}^{d \times K}$ are learnable parameters and $\mu_j \in \mathbb{R}^K$ is a learnable bias term. This formulation allows the gating to depend on the context provided by all features, relaxing the strict additivity of classical GAMs. To ensure that only the most relevant experts contribute to the final prediction, a sparsity-inducing mask $M_j \in \mathbb{R}^K$ is applied to φ_j, where entries corresponding to less significant scores are set to $-\infty$, with remaining values set to be 0. This masking procedure effectively zeroes out the contribution of less important experts after normalization. The relevance weights are then obtained via a softmax operation:

$$r_{jk} = \frac{\exp\big(\varphi_j[k] + M_j[k]\big)}{\sum_{l=1}^{K} \exp\big(\varphi_j[l] + M_j[l]\big)}, \quad k = 1, \cdots, K, \tag{4}$$

which ensures that $r_{jk} \geq 0$ and $\sum_{k=1}^{K} r_{jk} = 1$. Subsequently, the output for feature x_i is computed as a

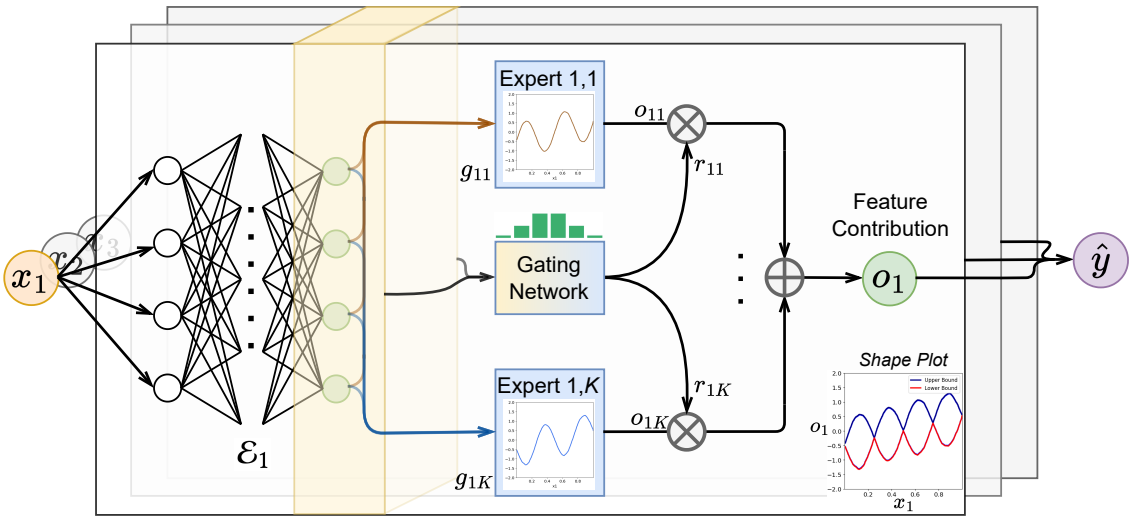


Figure 1: Illustration of the Neural Additive Expert (NAE) framework. A gating network dynamically assigns relevance scores to multiple expert predictors for each feature, and the aggregated feature contributions are summed to produce the final prediction. This design maintains interpretability while enabling flexible modeling of complex relationships.

weighted aggregation of the individual expert predictions:

$$o_i = \sum_{k=1}^{K} r_{ik} o_{ik}, \tag{5}$$

where o_{ik} is the output of the k-th expert for feature x_i as defined in Equation 2. The overall model prediction is obtained by summing the feature-specific outputs along with an intercept term ω_0:

$$\hat{y} = \omega_0 + \sum_{i=1}^{n} o_i. \tag{6}$$

This structure preserves feature-level interpretability while allowing for adaptive, context-dependent modeling. The computational complexity of NAEs is discussed in Appendix J.

2.3 Training Objective and Regularization

Given a dataset $\{(x^t, y^t)\}_{t=1}^{N}$, NAEs are trained to optimize:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{N}\sum_{t=1}^{N} \mathcal{L}(y^t, \hat{y}^t) + \\ & \frac{\lambda}{nNK} \sum_{t=1}^{N}\sum_{i=1}^{n}\sum_{k=1}^{K} \left(o_{ik}^t - \frac{1}{K}\sum_{l=1}^{K} o_{il}^t \right)^2, \end{aligned} \tag{7}$$

where $\mathcal{L}$ is a task-specific loss (e.g., mean squared error or cross-entropy), and λ controls the strength of the expert variation penalty. This regularization encourages consistency among experts for each feature, providing a tunable mechanism to balance flexibility and interpretability. Standard regularization techniques (e.g., dropout, L2 regularization) are also applied to promote generalization.

2.4 Interpretability and Feature Attribution

NAEs maintain interpretability by quantifying the contribution of each feature to the final prediction. For each feature x_i, the model aggregates expert outputs using the learned gating weights, ensuring that feature attributions remain bounded and meaningful. To further aid interpretation, we define upper and lower bounds for each feature's effect:

$$\begin{aligned} \text{upper}(o_i|x_i) &= \max_k g_{ik}\big(\mathcal{E}_i(x_i)\big), \\ \text{lower}(o_i|x_i) &= \min_k g_{ik}\big(\mathcal{E}_i(x_i)\big). \end{aligned} \tag{8}$$

These bounds characterize the range of possible contributions for each feature, and visualizing the distribution of actual feature effects within this range provides insight into both the magnitude and variability of feature influences.

While feature attribution is central to NAEs, the model also enables analysis of feature interactions. Since the score vector in Equation (3) is constructed as an additive function over all features, we can isolate the interaction between features x_i and x_j by focusing on their respective contributions in the gating function:

$$\varphi'_j = \mathcal{A}_{ij}^{\top} \mathcal{E}_i(x_i) \qquad \text{and} \qquad \varphi'_i = \mathcal{A}_{ji}^{\top} \mathcal{E}_j(x_j). \tag{9}$$

By removing the influence of other features, the updated feature outputs o'_i and o'_j with Equation (5) reflect the pairwise feature interactions captured by NAE. Further discussion of feature interactions is provided in Appendix F.

3 THEORETICAL ANALYSIS OF NAES

In this section, we characterize the expressivity of NAEs relative to classical additive families and analyze how they control model additivity for feature-level interpretability. Throughout we assume each feature domain $\mathcal{X}_i \subset \mathbb{R}$ is compact and all functions are continuous on their domains. NAEs use (i) per-feature encoders $E_i : \mathcal{X}_i \to \mathbb{R}^d$, (ii) expert functions $g_{ik} : \mathbb{R}^d \to \mathbb{R}$, (iii) softmax gating weights $r_{jk}(x) \geq 0$ with $\sum_{k=1}^{K} r_{jk}(x) = 1$, and (iv) per-feature outputs,

$$o_i(x) = \sum_{k=1}^{K} r_{ik}(x) g_{ik}\big(E_i(x_i)\big), \quad \hat{y}(x) = \omega_0 + \sum_{i=1}^{n} o_i(x),$$

as in Equations (2)-(6). Training uses the loss and expert-variation penalty from Equation (7). For theoretical clarity, we assume a dense expert routing architecture in which all K experts are available for each feature and gating is performed over the full set.

Model classes. Let GAM denote generalized additive models

$$\mathrm{GAM} = \Big\{ f(x) = \omega_0 + \sum_{i=1}^{n} f_i(x_i) \Big| f_i \text{ continuous} \Big\}. \quad (10)$$

Let GA^{2}M denote generalized additive models with pairwise interactions (GA^{2}M):

$$\mathrm{GA^2M} = \Big\{ f(x) = \omega_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{1 \leq i < j \leq n} f_{ij}(x_i, x_j) \Big| f_i, f_{ij} \text{ continuous} \Big\}. \quad (11)$$

We write NAE(K) for the set of functions represented by an NAE with K experts per feature.

3.1 Exact containment of GAMs

Theorem 1 (GAM containment)**.** For any $f \in \mathrm{GAM}$ there exists $K = 1$ and parameters of an NAE such that $\hat{y}(x) = f(x)$ for all x.

Proof. Choose $K = 1$, set $r_{i1}(x) \equiv 1$, and choose $g_{i1} \circ E_i = f_i$. Then $o_i(x) = f_i(x_i)$ and $\hat{y}(x) = \omega_0 + \sum_i f_i(x_i) = f(x)$. □

3.2 From GA^{2}M to NAE via separable approximation

To show the containment relative to GA^{2}M, we need the following approximation lemma, which follows from the Stone-Weierstrass theorem.

Lemma 1 (Finite separable approximation of pairwise terms)**.** Fix (i, j) and $\varepsilon > 0$. There exist $M \in \mathbb{N}$ and continuous functions $\{u_m : \mathcal{X}_i \to \mathbb{R},\ v_m : \mathcal{X}_j \to \mathbb{R}\}_{m=1}^{M}$ such that $\sup_{(x_i,x_j)} \big| f_{ij}(x_i, x_j) - \sum_{m=1}^{M} u_m(x_i) v_m(x_j) \big| < \varepsilon$.

We now show that each separable term $u(x_i)v(x_j)$ is representable by a single feature head $o_i(\cdot)$ using two experts and a softmax gate that depends on x_j only.

Lemma 2 (Two-expert product construction)**.** Let $u : \mathcal{X}_i \to \mathbb{R}$ and $v : \mathcal{X}_j \to \mathbb{R}$ be bounded and continuous. Choose $C > \sup_{x_j} |v(x_j)|$. Instantiate two experts for feature i with outputs $g_{i,+} \circ E_i(x_i) = +Cu(x_i)$, $g_{i,-} \circ E_i(x_i) = -Cu(x_i)$, and gate them by logits $\varphi_i^{(+)}(x) = \alpha(x) - \beta(x_j)$, $\varphi_i^{(-)}(x) = \alpha(x) + \beta(x_j)$ with softmax over $\{+, -\}$ only. Then

$$r_{i,+}(x) - r_{i,-}(x) = -\tanh\big(\beta(x_j)\big). \quad (12)$$

Setting $\beta(x_j) = -\operatorname{arctanh}\big(v(x_j)/C\big)$ yields

$$o_i(x) = r_{i,+}(x) C u(x_i) - r_{i,-}(x) C u(x_i) = u(x_i) v(x_j). \quad (13)$$

Moreover, we can enforce the "pairwise-only" dependence by choosing the gating parameters so that $\varphi_i^{(\pm)}$ depend on x_j but not on other coordinates (i.e., zeroing $A_{\ell i}$ for $\ell \neq j$ in Equation (3)).

Proof sketch. With softmax, if the logits are $\alpha \pm \beta$, the probabilities are $\sigma(-2\beta)$ and $\sigma(2\beta)$, hence their difference is $-\tanh(\beta)$. The rest is algebra using boundedness of v and the choice of C to keep $r_{i,\pm} \in [0, 1]$. □

Theorem 2 (GA^{2}M containment up to arbitrary precision)**.** For any $f \in \mathrm{GA^2M}$ and any $\epsilon > 0$ there exists K and an NAE such that $\sup_x |\hat{y}(x) - f(x)| < \epsilon$. In particular, for each feature i it suffices to take

$$K_i \leq 1 + 2 \sum_{j \neq i} M_{ij}, \quad (14)$$

where M_{ij} is the number of separable terms used for f_{ij} in Lemma 1.

Proof sketch. Write $f(x) = \omega_0 + \sum_i f_i(x_i) + \sum_{i<j} \sum_{m=1}^{M_{ij}} u_{ijm}(x_i) v_{ijm}(x_j) + r(x)$, with $\|r\|_\infty < \epsilon/2$ by Lemma 1 on each pair. Realize each f_i with a dedicated single expert (as in Theorem 1). For each separable $u_{ijm} v_{ijm}$ assign it to feature head i and apply Lemma 2. Summing all heads and the intercept recovers f up to the residual r, giving uniform error $< \epsilon$.

The bound on K_i counts one additive expert plus two per separable term. □

Thus, GAM $\subsetneq \overline{\text{NAE}(K)}$ (closure in uniform norm, i.e. functions uniformly approximable by NAE) whenever gates are allowed to depend on other features. Hence NAEs are more expressive than vanilla GAM/GA^{2}M but remain feature-decomposed at prediction time.

3.3 Controlling additivity via the expert-variation penalty

A key feature of NAEs is the ability to interpolate between strictly additive and highly flexible models by tuning the expert variation penalty. To quantify this, we define an additivity metric:

$$\text{Additivity} = \frac{1}{n}\sum_{i=1}^{n} \frac{\text{Var}\big(\mathbb{E}(o_i|x_i)\big) + \delta}{\text{Var}(o_i) + \delta}, \tag{15}$$

where o_i is the contribution of feature i to the prediction, and δ is a small constant for numerical stability. For a strictly additive model, this metric equals 1. The training objective of NAEs includes an expert variation penalty weighted by λ (Equation (7)), which encourages consistency among experts for each feature. As λ increases, the model becomes more additive:

Theorem 3 (Monotone additivity and additive limit)**.** Let $A(\lambda)$ be the additivity metric of an NAE trained with penalty $\lambda \geq 0$. Then $A(\lambda)$ is nondecreasing in λ. Moreover, $\lim_{\lambda\to\infty} A(\lambda) = 1$, and any limit point of minimizers as $\lambda \to \infty$ is a GAM, realized by setting $r_{ik} \equiv$ constants and g_{ik} equal across k for each feature.

Proof sketch. The penalty forces $g_{ik}(E_i(x_i))$ to contract toward their within-feature mean, collapsing each head to a single expert in the limit; gates then become irrelevant and the model reduces to an additive form, for which the metric equals 1. Monotonicity follows from standard convex-order arguments on the within-feature variance term. Full proofs are given in the supplement. □

4 EXPERIMENTS ON SIMULATED DATA

In this section, we empirically evaluate NAEs on synthetic datasets specifically constructed to assess their ability to model both additive and non-additive (multimodal) data distributions. We benchmark NAEs against traditional GAMs, implemented as Neural Additive Models (NAMs) (Agarwal et al., 2021). Our experiments address two main questions: (1) Can NAEs accurately recover underlying shape functions in both unimodal (additive) and multimodal (non-additive) scenarios? (2) How does the expert variation penalty parameter λ influence the trade-off between model flexibility and additivity? Additional simulation studies on robustness to feature sparsity and increasing distributional complexity are provided in Appendix E.

4.1 Simulation Setup and Visualization

We generate two synthetic datasets, each with 10,000 samples:

Unimodal Distribution (Additive). With $x_1 \sim U(0,1)$ and $\sigma = 0.1$, the response is sampled as:

$$y \sim \mathcal{N}\Big(x_1 - \frac{1}{2} + \sin(4\pi x_1), \sigma^2\Big), \tag{16}$$

Multimodal Distribution (Non-Additive). With x_2 uniformly sampled from $\{-1, 1\}$, the response is generated following:

$$y \sim \begin{cases} \mathcal{N}\Big(x_1 - \frac{1}{2} - \sin(4\pi x_1), \sigma^2\Big), & \text{if } x_2 = -1, \\ \mathcal{N}\Big(x_1 - \frac{1}{2} + \sin(4\pi x_1), \sigma^2\Big), & \text{if } x_2 = 1, \end{cases} \tag{17}$$

Figure 2 compares the learned shape functions of NAM and NAE. In the unimodal setting, both models accurately recover the underlying pattern, and NAE maintains interpretability without introducing spurious complexity. In the multimodal setting, NAM produces a near-linear fit, failing to capture the interaction, while NAE recovers the multimodal structure, with upper and lower expert bounds reflecting the range of feature contributions.

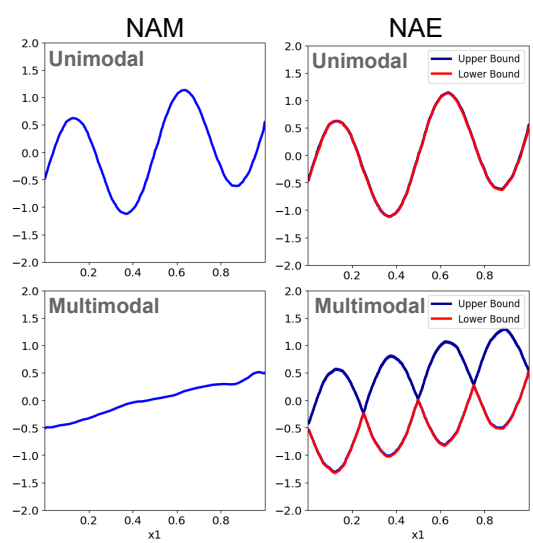


Figure 2: Shape functions learned by NAM and NAE on simulated data. For the multimodal case, NAE captures the oscillatory structure, while NAM fails.

Table 1: Performance comparison on benchmark datasets. "Complex." denotes the explanation complexity, i.e., the number of components required to fully explain model predictions, with n representing the number of features. The best-performing models in each category are highlighted.

Model	Interpretability	Complex.	Housing	MIMIC-II	MIMIC-III	Income	Credit	Year
			RMSE ↓	AUC ↑	AUC ↑	AUC ↑	AUC ↑	MSE ↓
Black-box Models								
MLP	No	N/A	0.501 ±0.006	0.835 ±0.014	0.815 ±0.009	0.914 ±0.003	**0.981 ±0.007**	78.48 ±0.56
NODE	No	N/A	0.523 ±0.000	0.843 ±0.011	**0.828 ±0.007**	0.919 ±0.003	0.981 ±0.009	**76.21 ±0.12**
XGBoost	No	N/A	**0.443 ±0.000**	**0.844 ±0.012**	0.819 ±0.004	**0.928 ±0.003**	0.978 ±0.009	78.53 ±0.09
Interaction-explaining Models								
EB^{2}M	Yes	$O(n^2)$	0.492 ±0.000	**0.848 ±0.012**	0.821 ±0.004	**0.928 ±0.003**	0.982 ±0.006	83.16 ±0.01
NA^{2}M	Yes	$O(n^2)$	0.492 ±0.008	0.843 ±0.012	**0.825 ±0.006**	0.912 ±0.003	0.985 ±0.007	79.80 ±0.05
NB^{2}M	Yes	$O(n^2)$	0.478±0.002	**0.848 ±0.012**	0.819 ±0.010	0.917 ±0.003	0.978 ±0.007	79.01 ±0.03
NODE-GA^{2}M	Yes	$O(n^2)$	**0.476 ±0.007**	0.846 ±0.011	0.822 ±0.007	0.923 ±0.003	**0.986 ±0.010**	**79.57 ±0.12**
Feature-explaining Models								
Linear	Yes	$O(n)$	0.735 ±0.000	0.796 ±0.012	0.772 ±0.009	0.900 ±0.002	0.976 ±0.012	88.89 ±0.40
Spline	Yes	$O(n)$	0.568 ±0.000	0.825 ±0.011	0.812 ±0.004	0.918 ±0.003	0.982 ±0.011	85.96 ±0.07
EBM	Yes	$O(n)$	0.559 ±0.000	0.835 ±0.011	0.809 ±0.004	**0.927 ±0.003**	0.974 ±0.009	85.81 ±0.11
NAM	Yes	$O(n)$	0.572 ±0.005	0.834 ±0.013	0.813 ±0.003	0.910 ±0.003	0.977 ±0.015	85.25 ±0.01
NBM	Yes	$O(n)$	0.564 ±0.001	0.833 ±0.013	0.806 ±0.003	0.918 ±0.003	0.981 ±0.007	85.10 ±0.01
NODE-GAM	Yes	$O(n)$	0.558 ±0.003	0.832 ±0.011	0.814 ±0.005	**0.927 ±0.003**	0.981 ±0.011	85.09 ±0.01
NAE	Yes	$O(n)$	**0.451 ±0.002**	**0.847 ±0.014**	**0.825 ±0.006**	**0.927 ±0.003**	**0.982 ±0.009**	**78.66 ±0.21**

4.2 Effect of Variation Penalty λ

The expert variation penalty λ in NAEs controls the balance between flexibility and additivity, as shown in our theoretical analysis (Section 3). To further validate this, we train NAEs on the multimodal simulated dataset with different regularization strengths $\lambda = 0.1, 1, 10$ and visualize the learned shape functions for x_1.

As shown in Figure 3, small λ values allow for high flexibility, which perfectly recovers the underlying distribution and has a wide gap between upper and lower bounds. As λ increases, the bounds converge, enforcing additivity of the trained model. The corresponding additivity scores (0.597, 0.709, and 1.000 for $\lambda = 0.1, 1, 10$) confirm the theoretical predictions.

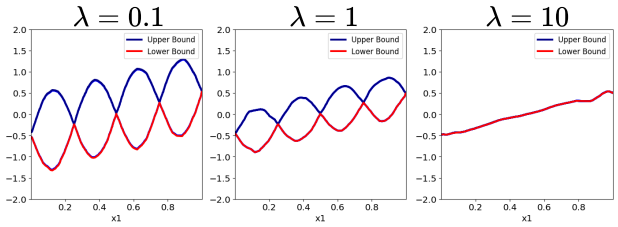


Figure 3: Effect of the expert variation penalty λ on the learned shape function of x_1 in NAEs. As λ increases, the model transitions from flexible to strictly additive.

5 EXPERIMENTS ON REAL-WORLD DATA

In this section, we evaluate NAEs on a suite of real-world datasets, focusing on both predictive accuracy and interpretability. We describe the datasets and baseline models, analyze predictive performance and feature attributions, and investigate the trade-off between accuracy and interpretability. Additional ablation studies on the gating mechanism and number of experts are provided in Appendices G, H, and I.

5.1 Datasets and Baselines

We evaluate NAEs on six widely used datasets spanning regression and classification tasks: Housing (Pace and Barry, 1997), MIMIC-II (Saeed et al., 2011), MIMIC-III (Johnson et al., 2016), Income (Blake, 1998), Credit, and Year. These datasets vary in size, feature types, and the presence of categorical variables, providing a comprehensive testbed. Dataset statistics and more details are in Appendix A.

We benchmark NAEs against a range of baselines: linear and spline models (basic interpretability), Neural Additive Models (NAM) (Agarwal et al., 2021), Neural Basis Models (NBM) (Radenovic et al., 2022), Explainable Boosting Machine (EBM) (Lou et al., 2012; Nori et al., 2019), NODE-GAM (Chang et al., 2022), and their pairwise-interaction variants (e.g., NA^{2}M, NB^{2}M, NODE-GA^{2}M). Black-box models such as

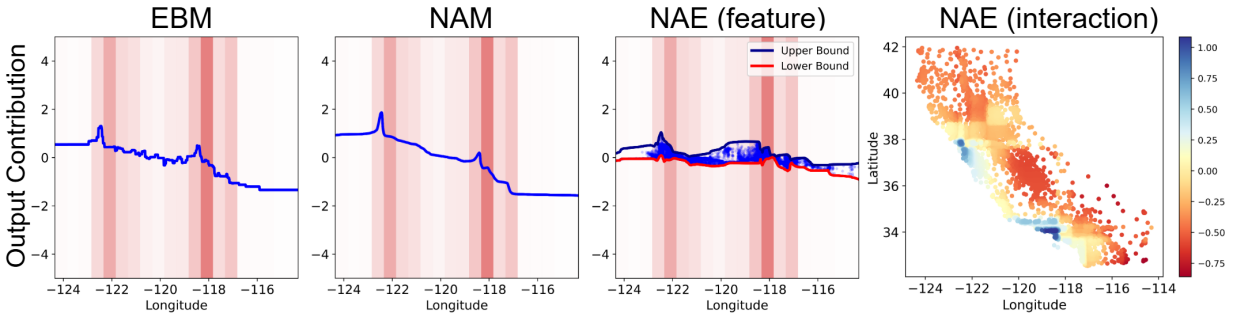


Figure 4: Comparison of Longitude effect on house price predictions across models. Y-axis shows mean-centered feature contributions. Background bars indicate normalized data density. Blue dots show actual feature effects. Longitude-Latitude interaction learned by NAE is also shown.

MLP, NODE (Popov et al., 2020), and XGBoost (Chen and Guestrin, 2016) are also included. Implementation details are in Appendices B and C.

5.2 Predictive Performance and Feature Attribution

Table 1 compares NAEs to all baselines. Metrics are reported with arrows indicating the desired direction. The results show that while traditional additive models (e.g., EBM, NAM) provide feature-level explanations, they often underperform compared to models that capture interactions or use black-box architectures. NAEs, however, match or exceed the predictive performance of these more complex models, while maintaining clear, feature-level interpretability.

The "Complex." column in the table refers to the number of components required for a full model inspection. For interaction-level GA^{2}M models, the prediction sums univariate functions and pairwise interaction surfaces, meaning a complete inspection requires examining $O(n^2)$ components. In contrast, an NAE prediction always decomposes into exactly one scalar contribution per feature. The primary explanation thus consists of only $O(n)$ shape plots, each displaying the feature's main effect and its associated interaction strength bounds. In high-dimensional settings, this reduction from $O(n^2)$ to $O(n)$ explanation complexity offers a significant interpretability advantage.

Figure 4 demonstrates the interpretability of NAEs by comparing the effect of the Longitude feature on house price predictions across models. EBM, NAM, and NAE all capture similar geographic trends, with higher property values near key longitudes (e.g., -122.5 for San Francisco, -118.5 for Los Angeles). NAE further reveals a wide range of possible outcomes between -120 and -119, with most data points near the lower bound. Its Longitude-Latitude interaction plot confirms that coastal regions positively influence house prices, while other areas in the same longitude range may have a negative effect.

5.3 Controlling the Trade-off Between Accuracy and Interpretability

As demonstrated by the theoretical analysis and experiments on the simulated data, NAEs provide a tunable trade-off between flexibility and additivity via the expert variation penalty parameter λ. Figure 5 shows shape plots for the Longitude feature with different λ values. As λ increases, the bounds on feature effects tighten, enforcing additivity but potentially reducing flexibility.

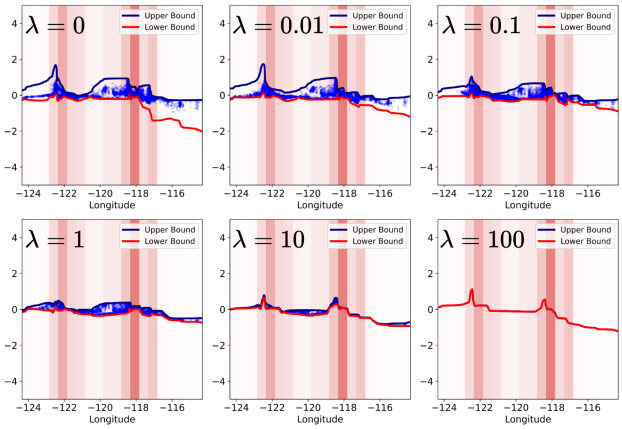


Figure 5: Shape plots of the Longitude feature learned by NAEs with different λ values. Higher λ enforces additivity and narrows the range of possible feature effects.

To quantify this trade-off, we report both the additivity metric from Equation (15) and the RMSE of the trained NAEs. Additionally, we introduce a tightness

metric to assess how closely the predicted bounds capture the estimated feature effects, reflecting the variability in the model's predictions. The tightness metric is defined as follows:

$$\text{Tightness} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\Big[\frac{\max(o_i|x_i) - \min(o_i|x_i) + \delta}{\text{upper}(o_i|x_i) - \text{lower}(o_i|x_i) + \delta}\Big]. \tag{18}$$

A high tightness value indicates that the predicted values closely span the estimated bounds, while a low value suggests the bounds are overly loose.

Table 2: Quantitative analysis of additivity and bound tightness on the Housing dataset for various λ settings, along with corresponding performance (RMSE).

$\lambda = \mathbf{0}$	$\lambda = \mathbf{0.01}$	$\lambda = \mathbf{0.1}$	$\lambda = \mathbf{1}$	$\lambda = \mathbf{10}$	$\lambda = \mathbf{100}$
Additivity					
0.522 ±0.000	0.537 ±0.000	0.562 ±0.012	0.639 ±0.000	0.897 ±0.000	1.000 ±0.000
Tightness					
0.860 ±0.000	0.895 ±0.018	0.953 ±0.011	0.988 ±0.000	0.999 ±0.000	1.000 ±0.000
RMSE					
0.451 ±0.003	0.450 ±0.002	0.451 ±0.003	0.458 ±0.003	0.515 ±0.008	0.582 ±0.008

Table 2 summarizes additivity, tightness, and RMSE on the Housing dataset for different λ values. As λ increases, additivity and tightness improve, but predictive performance (RMSE) may degrade. This confirms that λ provides a practical mechanism for balancing accuracy and interpretability in NAEs. In our main experiments, we use $\lambda = 0.1$ uniformly across datasets, as it offers a good balance between flexibility and additivity. For practical use, we suggest starting with $\lambda = 0.1$ and increasing it if greater interpretability is needed, monitoring both validation accuracy and the additivity metric.

6 RELATED WORK

6.1 Generalized Additive Models

Generalized Additive Models (GAMs) are foundational in interpretable machine learning, modeling the response as a sum of univariate functions for clear, feature-level attributions (Hastie, 2017). This additive structure enables visualization and understanding of individual feature effects, making GAMs valuable in transparency-critical domains. Extensions include hierarchical modeling (Pedersen et al., 2019), robust evaluation (Hegselmann et al., 2020; Chang et al., 2021), and higher-order interactions (Enouen and Liu, 2022; Tan et al., 2018; Duong et al., 2024). Recent advances such as Explainable Boosting Machines (EBMs) (Lou et al., 2012), Neural Additive Models (NAMs) (Agarwal et al., 2021), NODE-GAM (Chang et al., 2022), and ProtoNAM (Xiong et al., 2025) improve flexibility and accuracy while preserving interpretability. Further developments, including Neural Basis Models (NBMs) (Radenovic et al., 2022) and Gaussian Process Neural Additive Models (GP-NAMs) (Zhang et al., 2024), enhance parameter efficiency, scalability, and uncertainty quantification (Bouchiat et al., 2023; Thielmann et al., 2024). However, the strictly additive nature of GAMs limits their ability to capture complex, context-dependent feature interactions (Cui et al., 2020; Mueller et al., 2024; Ibrahim et al., 2023).

Previous work has attempted to address this limitation by introducing models such as GA^{2}M, where interactions are also captured (Lou et al., 2013). GA^{2}M primarily relies on 2D interaction plots for interpretability, which can become computationally expensive and challenging to visualize as the number of features increases. Also, multiple interaction plots are often required to fully understand the effect of a single feature in GA^{2}M, which complicates interpretation. While post-hoc heterogeneity analysis methods were proposed to estimate the single feature effect in GA^{2}M without visualizing all interactions (Goldstein et al., 2015; Gkolemis et al., 2023), their explanations are empirical variability bands based on given samples. Although such bands can be informative, they do not guarantee coverage for unseen samples. In contrast, NAEs provide architectural (model-intrinsic) bounds on feature effects defined by the minimum and maximum outputs of the learned expert functions, ensuring coverage for any context and input, including those not seen during training.

Beyond GAMs, there are also post-hoc additive explanation methods like additive SHAP (Lundberg and Lee, 2017), which decompose predictions of complex models into additive contributions. However, these methods provide approximations of the underlying black-box model's behavior rather than inherently interpretable architectures, and they can be computationally expensive for large models. In contrast, NAEs build additivity directly into the architecture and training objective. Because the model is defined as a sum of per-feature expert contributions, the resulting explanations are exact by construction rather than approximations of a more complex underlying function. Furthermore, unlike post-hoc methods that can also be applied to trained models, NAEs allow explicit control over the models during training via the expert-variation penalty λ. This parameter governs the deviation from strict additivity, enabling a direct trade-

off between expressivity and interpretability. Additionally, NAE feature attributions are inherent to the forward pass, avoiding the computational overhead of estimating SHAP values for complex models.

6.2 Mixture of Experts

The Mixture of Experts (MoE) paradigm introduces a modular approach to modeling, where multiple specialized sub-models (experts) are trained and a gating network dynamically selects or combines their outputs (Jacobs et al., 1991). MoE architectures have demonstrated remarkable scalability and adaptability, particularly in large-scale neural networks. Notably, sparse gating mechanisms have enabled efficient training of massive models by activating only a subset of experts per input (Shazeer et al., 2017). MoE has been successfully integrated into Transformer architectures (Vaswani et al., 2017), powering advances in large language models (Artetxe et al., 2022; Jiang et al., 2024; Lepikhin et al., 2021) and multimodal systems (Lin et al., 2024; Shen et al., 2023).

While MoE models excel at capturing diverse patterns and scaling to large datasets, they typically lack the interpretability of additive models. Specifically, standard MoE architectures employ experts that operate on the full input vector, with a gating network combining their outputs into a single prediction. While flexible, this approach sacrifices the decomposability required for interpretability. In contrast, NAEs organize experts per feature, ensuring that the final prediction remains a sum of feature-wise contributions (Equation (6)) even when the gating network utilizes all features. Similarly, unlike general modular networks which focus on functional specialization without enforcing additive constraints, NAEs enforce modularization specifically at the feature level. This design, combined with our expert-variation penalty, allows NAEs to function as a structured, interpretable variant of MoE that retains GAM-like additivity while offering explicit control over interaction strength.

6.3 Integrating Additive Models with Mixture of Experts

Several efforts have sought to combine the interpretability of GAMs with the flexibility of MoE architectures, but important limitations remain. The Generalized Additive Models for Location, Scale, and Shape (GAMLSS) framework (Rigby and Stasinopoulos, 2005) extends GAMs to model multiple distributional parameters (e.g., mean and variance) in an additive manner, increasing flexibility but still failing to capture complex feature interactions. FlexMix (Leisch, 2004) introduces a mixture model of GAMs with expert weights estimated via Expectation-Maximization, yet these weights are fixed after training and cannot adapt to new inputs, limiting expressiveness. Mixdistreg (Rügamer, 2023; Rügamer et al., 2024) and MNAM (Kim et al., 2024) propose a dynamic mixture of additive models with input-dependent weights, but apply the same set of weights uniformly across all features or experts, preventing feature-specific adaptation and lacking explicit mechanisms to balance interpretability and accuracy.

In summary, while prior work has made progress toward integrating interpretability and accuracy, existing methods either retain strict additivity or lack fine-grained control over the trade-off between accuracy and transparency. This motivates the development of new frameworks, such as NAEs, that provide both feature-level interpretability and adaptive modeling capacity. By enabling context-dependent expert selection and feature-specific adaptation, our model bridges the gap between transparent, interpretable models and highly flexible, accurate architectures, opening new possibilities for trustworthy machine learning in complex domains.

7 CONCLUSION

We introduced Neural Additive Experts (NAEs), a novel framework that balances interpretability and predictive accuracy by extending additive models with a mixture-of-experts architecture and dynamic gating. NAEs enable each feature to be modeled by multiple specialized networks, with a gating mechanism that adaptively integrates information across features, thus relaxing strict additive constraints. Our theoretical analysis demonstrates that NAEs can recover complex, context-dependent relationships that are inaccessible to standard GAMs, while targeted regularization provides explicit control over the trade-off between flexibility and transparency. Empirical results on both synthetic and real-world datasets show that NAEs achieve competitive accuracy with state-of-the-art models, while maintaining clear, feature-level explanations. This work provides a principled and practical approach for applications where both interpretability and accuracy are essential.

Acknowledgements

This work is supported in part by the US National Science Foundation (NSF) and the National Institute of Health (NIH) under grants IIS-2106913, IIS-2538206, IIS-2529378, CCF-2217071, CNS-2213700, and R01LM014012-01A1. Any recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIH or NSF.

References

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. (2021). Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711.

Artetxe, M., Bhosale, S., Goyal, N., Mihaylov, T., Ott, M., Shleifer, S., Lin, X. V., Du, J., Iyer, S., Pasunuru, R., Anantharaman, G., Li, X., Chen, S., Akin, H., Baines, M., Martin, L., Zhou, X., Koura, P. S., O'Horo, B., Wang, J., Zettlemoyer, L., Diab, M., Kozareva, Z., and Stoyanov, V. (2022). Efficient large scale language modeling with mixtures of experts. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Biran, O. and Cotton, C. (2017). Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, pages 8–13.

Blake, C. (1998). Uci repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/ML-Repository. html*.

Bouchiat, K., Immer, A., Yèche, H., Rätsch, G., and Fortuin, V. (2023). Improving neural additive models with bayesian principles.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730.

Chang, C.-H., Caruana, R., and Goldenberg, A. (2022). NODE-GAM: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*.

Chang, C.-H., Tan, S., Lengerich, B., Goldenberg, A., and Caruana, R. (2021). How interpretable and trustworthy are gams? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 95–105.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Christoph, M. (2020). Interpretable machine learning: A guide for making black box models explainable.

Cui, Z., Fritz, B. A., King, C. R., Avidan, M. S., and Chen, Y. (2020). A factored generalized additive model for clinical decision support in the operating room. In *AMIA Annual Symposium Proceedings*, volume 2019, page 343.

Duong, V., Wu, Q., Zhou, Z., Zhao, H., Luo, C., Zavesky, E., Yao, H., and Shao, H. (2024). Cat: Interpretable concept-based taylor additive models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 723–734.

Enouen, J. and Liu, Y. (2022). Sparse interaction additive networks via feature interaction detection and sparse selection. *Advances in Neural Information Processing Systems*, 35:13908–13920.

Gkolemis, V., Dalamagas, T., Ntoutsi, E., and Diou, C. (2023). Rhale: Robust and heterogeneity-aware accumulated local effects. In *ECAI*, pages 859–866.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Hegselmann, S., Volkert, T., Ohlenburg, H., Gottschalk, A., Dugas, M., and Ertmer, C. (2020). An evaluation of the doctor-interpretability of generalized additive models with interactions. In *Machine Learning for Healthcare Conference*, pages 46–79. PMLR.

Ibrahim, S., Afriat, G., Behdin, K., and Mazumder, R. (2023). Grand-slamin'interpretable additive modeling with structural constraints. *Advances in Neural Information Processing Systems*, 36:61158–61186.

Ibrahim, S., Radchenko, P., Ben-David, E., and Mazumder, R. (2025). Predicting census survey response rates with parsimonious additive models and structured interactions. *The Annals of Applied Statistics*, 19(1):94–120.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Kim, Y. K., Di Martino, J. M., and Sapiro, G. (2024). Generalizing neural additive models via statistical multimodal analysis. *Transactions on Machine Learning Research*.

Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent glass regression in r.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. (2021). {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Zhang, J., Ning, M., and Yuan, L. (2024). Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lou, Y., Caruana, R., and Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158.

Lou, Y., Caruana, R., Gehrke, J., and Hooker, G. (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

McIntosh, C. N. (2025). Generalized additive logit models for clinical prediction. *Blood Advances*, 9(4):935–936.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Mueller, A., Siems, J., Nori, H., Salinas, D., Zela, A., Caruana, R., and Hutter, F. (2024). Gamformer: In-context learning for generalized additive models. *arXiv preprint arXiv:2410.04560*.

Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Pace, R. K. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297.

Pedersen, E. J., Miller, D. L., Simpson, G. L., and Ross, N. (2019). Hierarchical generalized additive models in ecology: an introduction with mgcv. *PeerJ*, 7:e6876.

Popov, S., Morozov, S., and Babenko, A. (2020). Neural oblivious decision ensembles for deep learning on tabular data. In *International Conference on Learning Representations*.

Radenovic, F., Dubey, A., and Mahajan, D. (2022). Neural basis models for interpretability. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8414–8426. Curran Associates, Inc.

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 54(3):507–554.

Rügamer, D. (2023). mixdistreg: An r package for fitting mixture of experts distributional regression with adaptive first-order methods. *arXiv preprint arXiv:2302.02043*.

Rügamer, D., Pfisterer, F., Bischl, B., and Grün, B. (2024). Mixture of experts distributional regression: implementation using robust estimation with adaptive first-order methods. *AStA Advances in Statistical Analysis*, 108(2):351–373.

Ruppert, D. (2004). The elements of statistical learning: data mining, inference, and prediction.

Saeed, M., Villarroel, M., Reisner, A. T., Clifford, G., Lehman, L.-W., Moody, G., Heldt, T., Kyaw, T. H., Moody, B., and Mark, R. G. (2011). Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Shen, S., Yao, Z., Li, C., Darrell, T., Keutzer, K., and He, Y. (2023). Scaling vision-language models with sparse mixture of experts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11329–11344.

Tan, S., Caruana, R., Hooker, G., Koch, P., and Gordo, A. (2018). Learning global additive explanations for neural nets using model distillation. *stat*, 1050:3.

Thielmann, A. F., Kruse, R.-M., Kneib, T., and Säfken, B. (2024). Neural additive models for location scale and shape: A framework for interpretable neural regression beyond the mean. In *International Conference on Artificial Intelligence and Statistics*, pages 1783–1791. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Xiong, G., Sinha, S., and Zhang, A. (2025). Protonam: Prototypical neural additive models for interpretable deep tabular learning. *ACM Trans. Knowl. Discov. Data*, 19(9).

Zhang, W., Barr, B., and Paisley, J. (2024). Gaussian process neural additive models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16865–16872.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Context-Gated Experts for Controllable Model Additivity: Supplementary Materials

A DATASET DESCRIPTION

Housing Dataset The California Housing dataset, cited from Pace and Barry (1997), encompasses a regression task based on median housing prices across California's census blocks. It comprises 20,640 instances, each characterized by eight distinct attributes.

MIMIC-II Dataset The MIMIC-II (Multiparameter Intelligent Monitoring in Intensive Care) dataset, as described by Saeed et al. (2011), facilitates a binary classification task aimed at predicting mortality in intensive care units (ICUs). It contains 17 attributes, including seven categorical variables.

MIMIC-III Dataset MIMIC-III is a more extensive and detailed iteration of the MIMIC database referenced from Johnson et al. (2016). In our study, the same setting in NODE-GAM (Chang et al., 2022) is adopted, wherein categorical variables have been transformed into dummy variables for enhanced analysis.

Income Dataset Originating from the UCI Machine Learning Repository (Blake, 1998), the Income dataset underpins a binary classification task. Its objective is to predict whether an individual earns more than $50,000 annually.

Credit Dataset The Credit dataset[1] provides samples for a binary classification task on transaction fraud detection. It contains 30 anonymized features including 28 coefficients of PCA components. In Credit, 492 out of 284,807 transactions are labeled as frauds.

Year Dataset The Year dataset[2] contains data for a regression task, which uses the audio features to predict the release year of a song. It includes 515,345 samples with 90 features.

The statistics of each dataset can be found in Table 3.

Table 3: Statistics of real-world datasets used in model evaluation.

Dataset	Task	#Instances	#Features	#Categorical Features	#Classes
Housing	regression	20,640	8	0	–
MIMIC-II	classification	24,508	17	7	2
MIMIC-III	classification	27,348	57	0	2
Income	classification	32,561	14	8	2
Credit	classification	284,807	30	0	2
Year	regression	515,345	90	0	–

B IMPLEMENTATION DETAILS

For a comprehensive and equitable evaluation, we employed the officially released codes for baseline models including NODE, NODE-GAM/NODE-GA^{2}M, EBM/EB^{2}M, and NBM/NB^{2}M. We adopted the PyTorch implementation of NAM/NA^{2}M by NBM, which has been benchmarked against their model, for a direct comparison

[1]The Credit dataset can be downloaded at `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`.
[2]The Year dataset can be downloaded at `https://archive.ics.uci.edu/dataset/203/yearpredictionmsd`

with our proposed models. In developing NAEs, we constructed a multi-channel, fully connected network capable of encoding multiple features independently in a parallel manner.

For a fair comparison with existing methods, our experiments employ the processed version of MIMIC-II, MIMIC-III, Income, and Credit from Chang et al. (2022), where the datasets are split into five parts for a five-fold cross-validation. The Housing dataset and the Year dataset are split into the training, validation, and test sets following setups of previous research (Chang et al., 2022; Radenovic et al., 2022), and each of them is tested with 10 different seeds to examine the variation of predictions. Consistent with previous studies (Chang et al., 2022; Popov et al., 2020; Radenovic et al., 2022), we applied the same quantile transformation to the features.

To be consistent with existing neural network-based GAMs (Agarwal et al., 2021; Radenovic et al., 2022), the feature encoders $f_1, \cdots, f_n$ in NAE are implemented with MLPs. The expert predictors $g_{11}, \cdots, g_{nK}$ are implemented as one linear layer in our experiments. During training, we decrease the learning rate with cosine annealing following NBM (Radenovic et al., 2022). AdamW optimizer (Loshchilov and Hutter, 2018) is used to optimize the training objective. All experiments were run on NVIDIA A100 GPUs (40 GB / 80 GB).

C HYPERPARAMETERS

Table 4: Hyperparameters for NAE on all datasets.

Hyperparameter	Housing	MIMIC-II	MIMIC-III	Income	Credit	Year
#layers	4	4	4	4	4	4
Hidden dimension	128	128	128	128	128	128
#total experts	4	4	4	4	4	4
#activated experts	4	4	4	4	4	4
Batch size	2048	2048	1024	2048	2048	512
Max iteration	1000	1000	500	1000	150	75
Learning rate	5.97e-4	1.97e-4	1.68e-4	1.11e-4	4.00e-5	1.17e-4
Weight decay	5.29e-5	8.06e-4	2.78e-4	4.25e-3	1.88e-3	2.28e-6
Dropout	0.1	0.0	0.1	0.0	0.3	0.1
Dropout expert	0.2	0.4	0.2	0.5	0.2	0.6
Output Penalty	1.97e-5	3.49e-8	4.86e-5	3.99e-1	5.07e-4	1.45e-4
Variation Penalty	0.1	0.1	0.1	0.1	0.1	0.1
Normalization	layer_norm	layer_norm	layer_norm	layer_norm	batch_norm	layer_norm

We perform random search over hyperparameters in the following ranges:

- #layers: the number of layers in each feature encoder, sampled from {3, 4}.
- Hidden dimension: the number of nodes in each layer of the feature encoder, sampled from {64, 128}.
- #total experts: the number of total experts for each feature, sampled from {4}.
- #activated experts: the number of activated experts for each feature, sampled from {4}.
- Batch size: the number of samples in one batch during training, sampled from {512, 1024, 2048}.
- Max iteration: the number of iterations for training, sampled from {75, 150, 500, 1000}.
- Learning rate: the speed of gradient descent, sampled from [1e-6, 1e-1].
- Weight decay: the coefficient for the L2 normalization on parameters, sampled from [1e-8, 1e-1].
- Dropout: the probability of a parameter in feature encoders being replaced as 0 during training, sampled from {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- Dropout expert: the probability of an expert's output being replaced as 0 during training, sampled from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- Output penalty: the coefficient for the L2 normalization on the outputs of features for reducing unimportant ones, sampled from [1e-8, 1e-1].

- Variation penalty: the weight for the expert variation penalty, sampled from {0.1}.
- Normalization: normalization methods used in the network, sampled from {batch_norm, layer_norm}.

The best hyperparameters we found for NAE are shown in Tables 4.

D PROOFS FOR SECTION 3

Throughout, each feature domain $\mathcal{X}_i$ is compact and metrizable, hence $\mathcal{X}_i$, $\mathcal{X}_i \times \mathcal{X}_j$ are compact Hausdorff. All functions are real-valued and continuous unless noted. We use $C(Y)$ for the Banach space of continuous real-valued functions on a compact space Y with the sup norm $\|\cdot\|_\infty$.

Lemma 3 (Restatement of Lemma 1)**.** Let $f_{ij} \in C(\mathcal{X}_i \times \mathcal{X}_j)$. For any $\varepsilon > 0$ there exist $M \in \mathbb{N}$ and functions $\{u_m \in C(\mathcal{X}_i),\, v_m \in C(\mathcal{X}_j)\}_{m=1}^{M}$ such that

$$\Big\| f_{ij} - \sum_{m=1}^{M} u_m(\cdot)\, v_m(\cdot) \Big\|_\infty < \varepsilon.$$

Proof. Consider the set

$$\mathcal{A} \;=\; \Big\{ \sum_{m=1}^{M} u_m(x_i)\, v_m(x_j) \;:\; M \in \mathbb{N},\; u_m \in C(\mathcal{X}_i),\; v_m \in C(\mathcal{X}_j) \Big\}.$$

$\mathcal{A}$ is a subalgebra of $C(\mathcal{X}_i \times \mathcal{X}_j)$: it is closed under pointwise addition and multiplication and contains constants (take $u \equiv 1$, $v \equiv 1$). It separates points: given $(x_i, x_j) \neq (x_i', x_j')$, either $x_i \neq x_i'$ or $x_j \neq x_j'$. If $x_i \neq x_i'$, choose $u \in C(\mathcal{X}_i)$ with $u(x_i) \neq u(x_i')$ and $v \equiv 1$. If $x_j \neq x_j'$, choose v analogously and $u \equiv 1$. By Stone–Weierstrass, the uniform closure of $\mathcal{A}$ is $C(\mathcal{X}_i \times \mathcal{X}_j)$. Thus every f_{ij} can be approximated arbitrarily well by finite sums of products $u_m v_m$. □

Lemma 4 (Restatement of Lemma 2)**.** Let $u \in C(\mathcal{X}_i)$ and $v \in C(\mathcal{X}_j)$ be bounded. Choose any constant $C > \|v\|_\infty$. Define two experts (for feature i)

$$g_{i,+} \circ E_i(x_i) = +C\, u(x_i), \quad g_{i,-} \circ E_i(x_i) = -C\, u(x_i),$$

and a two-class softmax gate with logits

$$\phi_i^{(+)}(x) = \alpha(x) - \beta(x_j), \quad \phi_i^{(-)}(x) = \alpha(x) + \beta(x_j).$$

Proof. With two-class softmax, if the logits are $\alpha \pm \beta$, the probabilities are

$$r_{i,+}(x) = \frac{1}{1 + e^{2\beta(x_j)}}, \quad r_{i,-}(x) = \frac{1}{1 + e^{-2\beta(x_j)}},$$

so $r_{i,+}(x) - r_{i,-}(x) = -\tanh(\beta(x_j))$. Setting $\beta(x_j) = -\operatorname{arctanh}(v(x_j)/C)$ yields $r_{i,+}(x) - r_{i,-}(x) = v(x_j)/C$. Then

$$o_i(x) = r_{i,+}(x)\, C\, u(x_i) + r_{i,-}(x)\, (-C\, u(x_i)) = u(x_i)\, v(x_j).$$

The dependence on x_j only is enforced by letting α be constant and β a function of x_j alone. □

Theorem 4 (Restatement of Theorem 1)**.** For any $f \in$ GAM, there exists an NAE with $K = 1$ per feature such that $\hat{y}(x) = f(x)$ for all x.

Proof. Set $K = 1$, $r_{i1}(x) \equiv 1$, choose $g_{i1} \circ E_i = f_i$ and ω_0 to match the intercept. Then $\hat{y}(x) = f(x)$. □

Theorem 5 (Restatement of Theorem 2)**.** Let $f \in \mathrm{GA}^2\mathrm{M}$. For any $\epsilon > 0$, there exists an NAE with context-only gating and finite experts $\{K_i\}$ such that $\|\hat{y} - f\|_\infty < \epsilon$. Moreover, one may take

$$K_i \;\leq\; 1 + 2 \sum_{j \neq i} M_{ij},$$

where M_{ij} is the number of separable terms used in Lemma 1.

Proof. Write $f(x) = \omega_0 + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j)$. Approximate each f_{ij} by $\sum_m u_{ijm}(x_i)v_{ijm}(x_j)$ with error $< \epsilon/(2\binom{n}{2})$. Realize each f_i via a single expert with constant gate. Realize each separable product $u_{ijm}(x_i)v_{ijm}(x_j)$ using Lemma 2. Summing all contributions gives $\hat{y}$ within ϵ. Counting experts yields the bound on K_i. □

Theorem 6 (Restatement of Theorem 3)**.** Let θ_λ minimize the penalized empirical objective in Equation (7):

$$\mathcal{L}_N(\theta;\lambda) = \frac{1}{N}\sum_{t=1}^{N} L\big(y_t, \hat{y}_\theta(x_t)\big) \;+\; \frac{\lambda}{nNK}\sum_{t=1}^{N}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big(o_{ik}(x_t) - \tfrac{1}{K}\sum_{\ell=1}^{K} o_{i\ell}(x_t)\Big)^2.$$

Then the within-feature dispersion $\mathsf{Pen}(\theta) := \frac{1}{nNK}\sum_{t,i,k}\big(o_{ik}(x_t) - \frac{1}{K}\sum_\ell o_{i\ell}(x_t)\big)^2$ is nonincreasing in λ, and $\mathsf{Pen}(\theta_\lambda) \to 0$ as $\lambda \to \infty$. Consequently, any limit point of $\hat{y}_{\theta_\lambda}$ is additive; moreover the additivity metric $A(\theta_\lambda)$ from Equation (15) satisfies $A(\theta_\lambda) \to 1$.

Proof. **(1) Monotonicity in λ.** Fix $0 \le \lambda_1 < \lambda_2$ and let $\theta_{\lambda_1}, \theta_{\lambda_2}$ be minimizers for λ_1, λ_2. By optimality,

$$\underbrace{\tfrac{1}{N}\sum_t L(y_t, \hat{y}_{\theta_{\lambda_2}}(x_t)) + \lambda_2\,\mathsf{Pen}(\theta_{\lambda_2})}_{\mathcal{L}_N(\theta_{\lambda_2};\lambda_2)} \;\le\; \underbrace{\tfrac{1}{N}\sum_t L(y_t, \hat{y}_{\theta_{\lambda_1}}(x_t)) + \lambda_2\,\mathsf{Pen}(\theta_{\lambda_1})}_{\mathcal{L}_N(\theta_{\lambda_1};\lambda_2)},$$

and

$$\mathcal{L}_N(\theta_{\lambda_1};\lambda_1) \;\le\; \mathcal{L}_N(\theta_{\lambda_2};\lambda_1) = \tfrac{1}{N}\sum_t L(y_t, \hat{y}_{\theta_{\lambda_2}}(x_t)) + \lambda_1\,\mathsf{Pen}(\theta_{\lambda_2}).$$

Subtracting the second inequality from the first yields $(\lambda_2 - \lambda_1)\mathsf{Pen}(\theta_{\lambda_2}) \le (\lambda_2 - \lambda_1)\mathsf{Pen}(\theta_{\lambda_1})$, so $\mathsf{Pen}(\theta_\lambda)$ is nonincreasing.

(2) $\mathsf{Pen}(\theta_\lambda) \to 0$. Take a zero-penalty competitor $\tilde{\theta}$ that implements a GAM with $K = 1$ (or $K > 1$ with $g_{ik} \equiv g_i$ so all o_{ik} are identical); this is feasible by Theorem 1 and Equations (2) - (6). Then for every λ,

$$\mathcal{L}_N(\theta_\lambda;\lambda) \;\le\; \mathcal{L}_N(\tilde{\theta};\lambda) \;=\; \tfrac{1}{N}\sum_t L\big(y_t, \hat{y}_{\tilde{\theta}}(x_t)\big) \;=: C.$$

Because the loss is nonnegative for the tasks considered (e.g., MSE, cross-entropy), we have $\lambda\,\mathsf{Pen}(\theta_\lambda) \le C$, hence $\mathsf{Pen}(\theta_\lambda) \le C/\lambda \to 0$.

(3) Collapse within each feature head at training points. Fix any feature i and training example x_t. By definition of Pen and Step (2),

$$\frac{1}{K}\sum_{k=1}^{K}\big(o_{ik}(x_t) - \bar{o}_i(x_t)\big)^2 \;\longrightarrow\; 0 \quad \text{where} \quad \bar{o}_i(x_t) = \tfrac{1}{K}\sum_{\ell=1}^{K} o_{i\ell}(x_t).$$

Thus the vector $\big(o_{i1}(x_t), \ldots, o_{iK}(x_t)\big)$ becomes constant in k: for each (i,t) there exists a scalar $q_i(x_t)$ with $o_{ik}(x_t) \to q_i(x_t)$ for all k. Since the feature contribution is $o_i(x_t) = \sum_k r_{ik}(x_t)\,o_{ik}(x_t)$ (Equation (5)), we get $o_i(x_t) \to \sum_k r_{ik}(x_t)\,q_i(x_t) = K\,q_i(x_t)$. Define $h_i(x_{i,t}) := K\,q_i(x_t)$; then

$$\hat{y}_{\theta_\lambda}(x_t) = \omega_0 + \sum_{i=1}^{n} o_i(x_t) \;\Longrightarrow\; \omega_0^\star + \sum_{i=1}^{n} h_i(x_{i,t}),$$

which is additive on the training set.

(4) Additivity in the limit and the metric $A(\lambda)$. At any *exact* zero-penalty parameter setting, $o_{ik}(x_t)$ is identical across k for every (i,t), so setting $g_{ik} \equiv g_i$ and any (e.g., constant) softmax weights r_{ik} yields a GAM realization with the same predictions (Equations (2) - (6)). For the additivity metric in Equation (15), when each o_i depends only on x_i, we have $\mathrm{Var}(\mathbb{E}[o_i \,|\, x_i]) = \mathrm{Var}(o_i)$, hence $A(\theta_\lambda) \to 1$ as $\lambda \to \infty$. □

E MORE SIMULATION STUDIES

E.1 Impact of Feature Sparsity

To assess robustness to imbalanced feature distributions, we simulate datasets where the proportion of instances with $x_2 = 1$ is set to 50%, 25%, 5%, and 1%. Figure 6 shows shape plots for traditional GAMs and NAEs with increasing numbers of experts ($K = 2, 8, 32$).

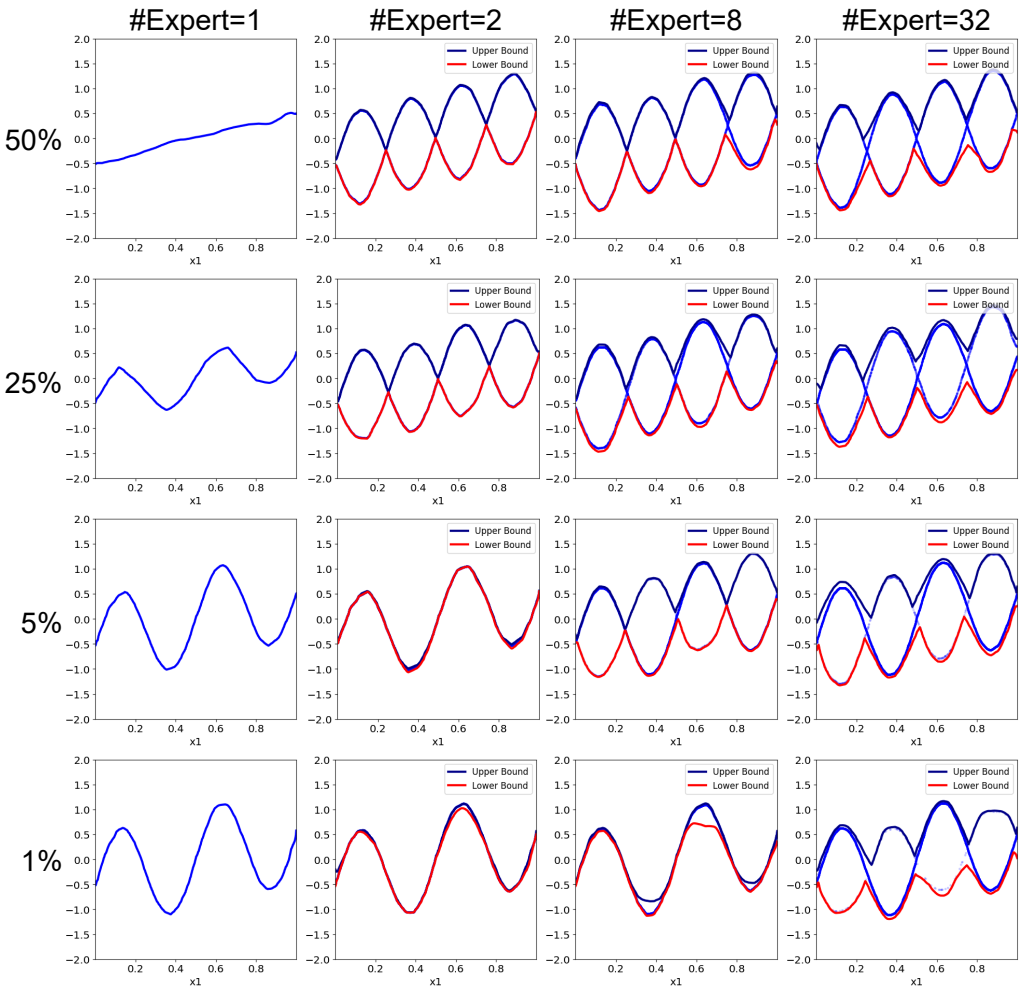


Figure 6: Shape plots of GAM and NAEs on simulated data with varying feature sparsity for x_2. NAEs recover minority signals even at high sparsity, while GAM overfits the majority class.

As sparsity increases, traditional GAMs overfit the majority class and miss minority patterns. NAEs, with more experts, consistently recover the multimodal structure, demonstrating robustness to feature imbalance.

E.2 Impact of Distribution Modality

We further test NAEs on data with increasing numbers of modes by introducing multiple categorical features:

$$y = \varepsilon + x_1 - \frac{1}{2} + \frac{1}{CF} \sum_{i=2}^{CF+1} x_i \sin(4\pi x_1), \tag{19}$$

where $\varepsilon \sim \mathcal{N}(0, 0.01^2)$, $x_1 \sim U(0, 1)$, and each x_i ($i \geq 2$) is uniformly sampled from $\{-1, +1\}$. We vary the number of categorical features ($CF = 1, 3, 5, 7$).

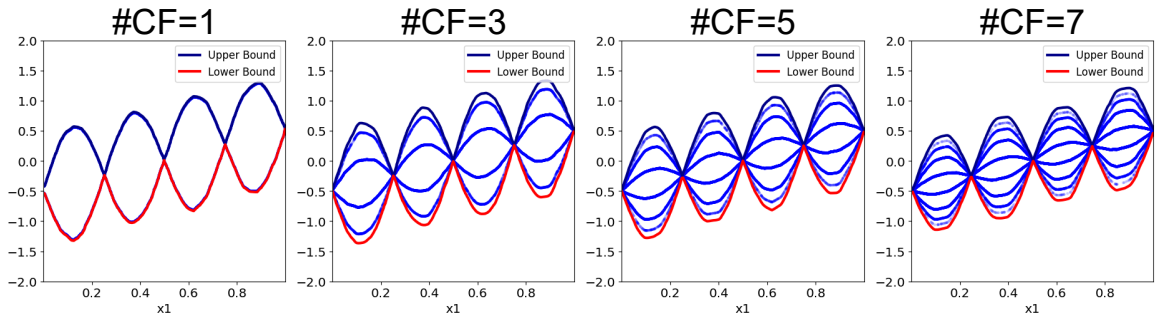


Figure 7: Shape plots of NAEs on simulated data with increasing numbers of categorical features. The model adapts to capture the growing complexity of the underlying function.

Figure 7 shows that NAEs flexibly capture the increasing complexity of the shape function as the number of modes grows, highlighting the model's scalability and adaptability to complex, multimodal distributions.

E.3 Impact of Correlated Features

To examine the effect of feature correlation, we simulate data where x_2 is correlated with x_1. Based on the multimodal setup in Equation (17), we introduce correlation between x_1 and x_2 by setting x_2 as a binary variable $\in \{-1, 1\}$ with conditional probability

$$P(x_2 = 1|x_1) = \rho x_1 + (1 - \rho)/2, \tag{20}$$

where $\rho \in \{0, 0.5, 0.9\}$ controls the correlation strength. The first feature x_1 is drawn uniformly from [0, 1], and the response is generated as

$$y = x_2 \sin(4\pi x_1) + x_2 + \varepsilon. \tag{21}$$

The comparison results (RMSE, lower is better) between NAE and NAM are shown in Table 5. As shown in the table, NAE maintains low RMSE across all correlation levels, whereas NAM performs significantly worse and only slightly improves with increased correlation. This demonstrates the robustness of NAE to correlated inputs.

Table 5: Performance comparison under different correlation levels.

Model	$\rho = 0.0$	$\rho = 0.5$	$\rho = 0.9$
NAE	0.1066	0.1068	0.1090
NAM	0.7112	0.6829	0.6088

E.4 Impact of Generic Interactions

To evaluate the ability of NAE to capture generic interactions, we simulate data with a complex interaction structure, where $x_1, x_2 \sim Uniform[-1, 1]$ and the response is generated as

$$y = 2\sin(\pi x_1)\cos(\pi x_2) + 0.5x_1^2 + 0.5x_2^2 + \varepsilon. \tag{22}$$

This target includes a multiplicative interaction term that cannot be decomposed additively. With the same evaluation settings as in previous simulations, we test both NAE and NAM on this data. The results show that NAE achieves a low RMSE of 0.1306, closely approximating the true function, while NAM fails to capture the interaction and yields a much higher RMSE of 1.0114. This illustrates that NAE can effectively learn complex interactions that are not strictly additive, as predicted by Theorem 2.

F COMPARISON BETWEEN NAE AND GA^{2}M

As demonstrated in Section 5, NAE provides both individual feature attributions and visualizations of feature interactions. To compare with GA^{2}M, which emphasizes feature interactions, we conducted additional experiments using EB^{2}M. Specifically, we obtained single-feature attributions by summing all interaction terms associated with each feature. For a fair comparison, we visualized the actual effect of each feature value, including the interaction effects, and plotted the lower and upper bounds (i.e., the minimum and maximum effect of a feature value across all captured interactions).

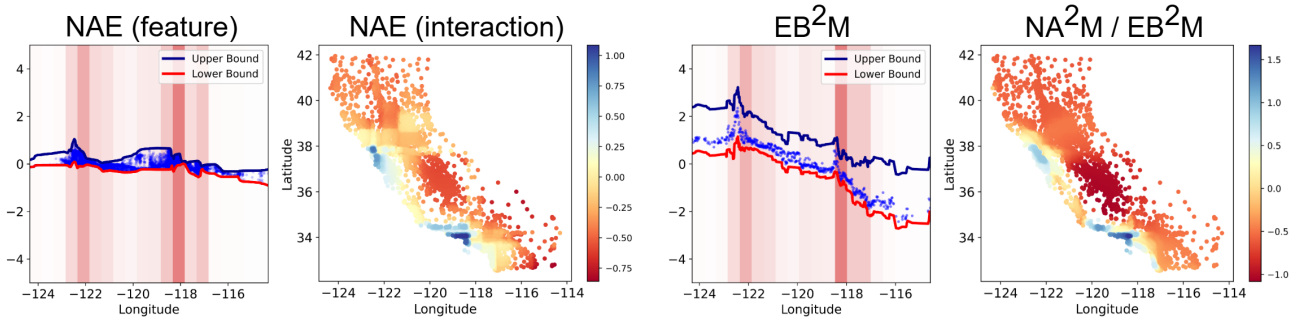


Figure 8: Comparison of NAE and GA^{2}M visualizations for the "Longitude" feature effect and the "Longitude-Latitude" interaction on house prices.

As shown in Figure 8, while the interaction interpretation by GA^{2}M is similar to NAE, the single-feature attributions from EB^{2}M yield bounds that are too loose to be interpretable, suggesting high variability in predictions, especially for out-of-distribution inputs. Also, this approach to visualizing individual feature attributions is not applicable to NA^{2}M, as it does not provide strict output ranges. Moreover, a key advantage of NAE is its ability to control model additivity, allowing a smooth transition from a complex interaction model to a strictly additive one, which is a flexibility not available in GA^{2}M.

Additionally, NAE is not limited to capturing pairwise interactions. For example, one can modify the relevance estimation in Equation (3) to keep the interaction terms among three features. Figure 9 focuses on the Los Angeles area, demonstrating the effect of "AveRooms" on the price prediction beyond the Latitude-Longitude interaction. Interestingly, the south-western part of the area is more robust to the AveRooms change compared to other parts, revealing a complex three-way interaction that GA^{2}M fails to capture.

G NAE-D: NAE WITH DIAGONAL ROUTING MATRIX

We explored a special version of NAE, termed NAE-D, where the scoring matrices compose a block-diagonal matrix. In NAE-D, the matrix $\mathcal{A}_{ij}$ in Equation (3) becomes a zero matrix if $i \neq j$. This configuration allows NAE-D to function similarly to GAMs, where the relevance estimation of a feature's experts is solely based on its own encoded information.

Since the relevance for experts of each feature is solely determined by itself in NAE-D, the estimated relevance scores will be fixed for the same feature value. To encourage the model to learn various distributions with multiple experts, we introduce randomness into the decision-making process by employing the Gumbel-softmax technique (Jang et al., 2016) with the temperature $\tau = 0.1$ to re-sample the experts during training based on their estimated relevance scores in Equation (4):

$$\hat{r}_{ik} = \exp\left(\frac{\log(r_{ik}) + g_i}{\tau}\right) \Big/ \sum_{l=1}^{K} \exp\left(\frac{\log(r_{il}) + g_l}{\tau}\right), \tag{23}$$

where

$$g_i, g_l \sim Gumbel(0, 1). \tag{24}$$

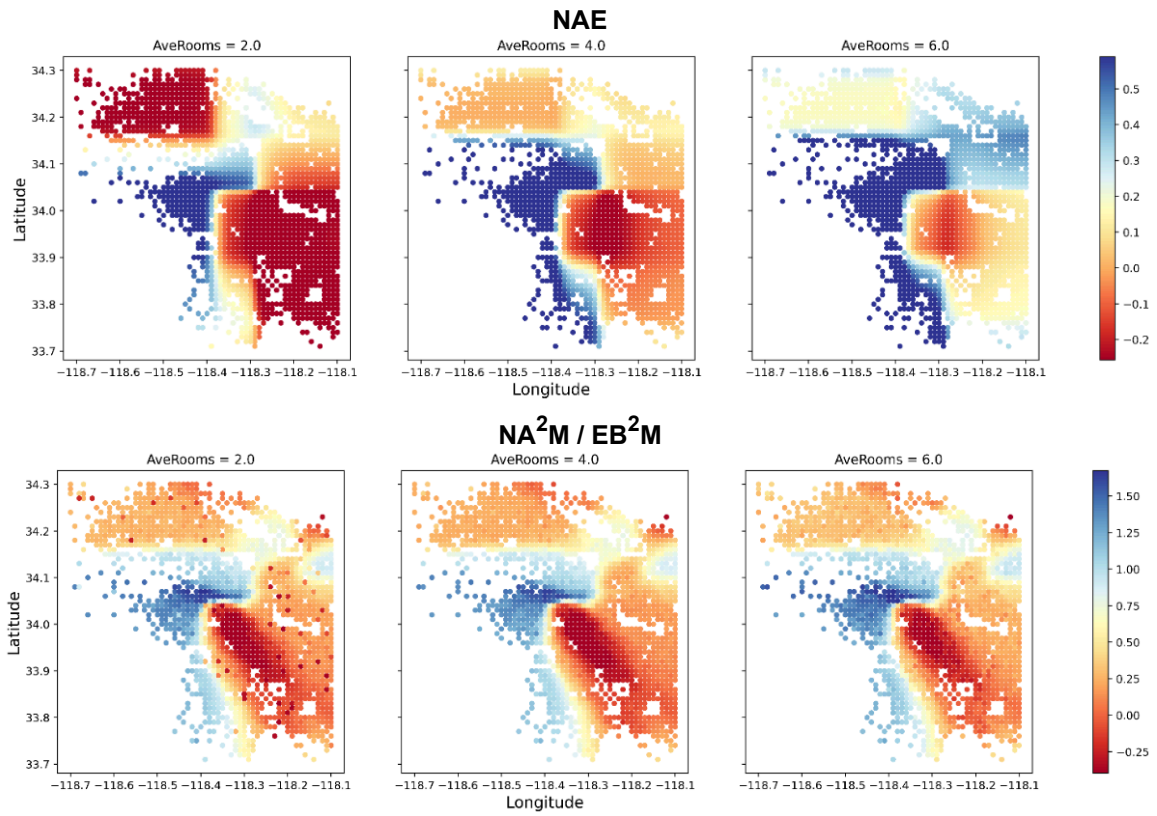


Figure 9: Comparison of NAE and GA^{2}M visualizations for the "AveRooms-Latitude-Longitude" interaction on house prices.

The performance comparison of NAE-D with other baselines is presented in Table 6. The results demonstrate that, by regularizing the scoring matrix to be diagonal, NAE-D shows a performance close to traditional additive models, which have worse accuracies than complex models that capture feature interactions.

Table 6: Comparison of NAE-D to other baselines on benchmark datasets. "Complex." denotes the explanation complexity, i.e., the number of components required to fully explain model predictions, with n representing the number of features.

Model	Interpret-ability	Complex.	Housing	MIMIC-II	MIMIC-III	Income	Credit	Year
			RMSE ↓	AUC ↑	AUC ↑	AUC ↑	AUC ↑	MSE ↓
Black-box Models								
MLP	No	N/A	0.501 ± 0.006	0.835 ± 0.014	0.815 ± 0.009	0.914 ± 0.003	0.981 ± 0.007	78.48 ± 0.56
XGBoost	No	N/A	0.443 ± 0.000	0.844 ± 0.012	0.819 ± 0.004	0.928 ± 0.003	0.978 ± 0.009	78.53 ± 0.09
Interaction-explaining Models								
EB^{2}M	Yes	$O(n^2)$	0.492 ± 0.000	0.848 ± 0.012	0.821 ± 0.004	0.928 ± 0.003	0.982 ± 0.006	83.16 ± 0.01
NA^{2}M	Yes	$O(n^2)$	0.492 ± 0.008	0.843 ± 0.012	0.825 ± 0.006	0.912 ± 0.003	0.985 ± 0.007	79.80 ± 0.05
Feature-explaining Models								
EBM	Yes	$O(n)$	0.559 ± 0.000	0.835 ± 0.011	0.809 ± 0.004	0.927 ± 0.003	0.974 ± 0.009	85.81 ± 0.11
NAM	Yes	$O(n)$	0.572 ± 0.005	0.834 ± 0.013	0.813 ± 0.003	0.910 ± 0.003	0.977 ± 0.015	85.25 ± 0.01
NAE-D	Yes	$O(n)$	0.553 ± 0.001	0.830 ± 0.012	0.805 ± 0.006	0.927 ± 0.003	0.977 ± 0.010	85.60 ± 0.04

Figure 10 shows the shape plots generated by NAE-D compared to other baseline additive models. As discussed above, NAE-D is a strictly additive model where the effect of each feature on the final output is determined

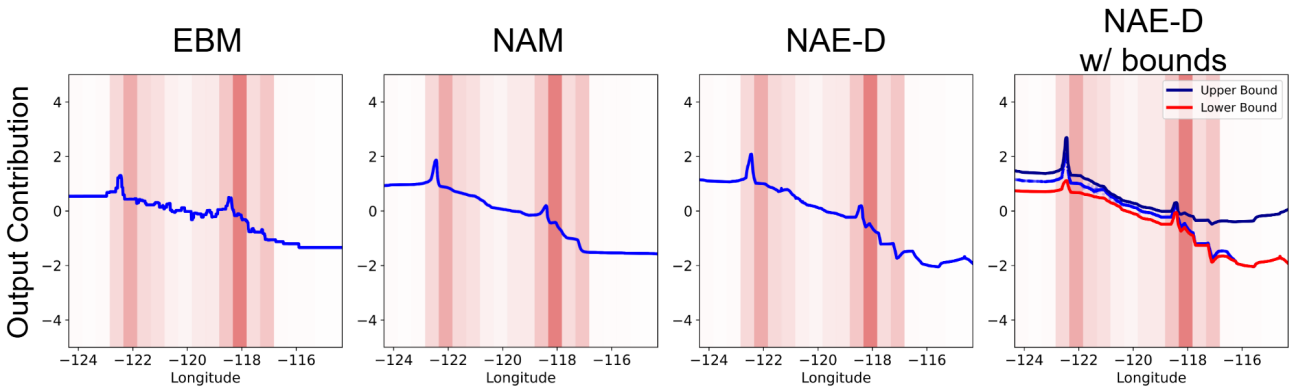


Figure 10: Visualization of the "Longitude" feature impact on house prices by NAE-D.

by the feature itself. Thus, we plot NAE-D in the same way as we did for EBM and NAM, directly showing the estimated contribution given by each feature value. Moreover, since NAE-D is trained under the general NAE framework with multiple experts learned for each feature, we plot another figure for NAE-D including the estimated upper and lower bounds based on the learned experts.

It can be observed from Figure 10 that the patterns captured by NAE-D are similar to the ones by EBM and NAM. Beyond the point estimation provided by traditional additive models, NAE-D offers additional insights with its prediction bounds. In areas with low data density, the plot displays a wide gap between the upper and lower bounds, highlighting variability in predictions where less data is available.

H NAE-E: NAE WITH EVENLY DISTRIBUTED EXPERT ACTIVATION

While the activation strategy detailed in Section 2.2 facilitates continuous expert relevance estimation, we propose an adaptation that shifts this estimation to a discrete framework. This modification involves adjusting the weights to be evenly distributed across relevant experts, leading to a modified model termed NAE-E. To implement this, we use the masking vector outlined in Equation (3) and modify the relevance computation in Equation (4) as follows:

$$r_{ik} = \frac{\exp(\varphi_i[k] - \varphi_i^*[k] + M_i[k])}{\sum_{l=1}^{K} \exp(\varphi_i[l] - \varphi_i^*[l] + M_i[l])}. \tag{25}$$

φ_i^* mirrors the values of φ_i but does not require gradient computations. The activation strategy implemented by Equation (25) will result in a finite set of possible outcomes for each input feature value. Given a configuration of K experts and C activated ones per feature, the predicted output by NAE-E for any given feature will be one of the possible selections from $\frac{K!}{C!(K-C)!}$ combinations.

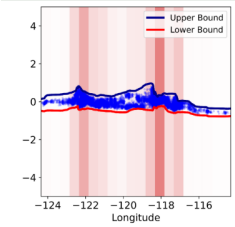
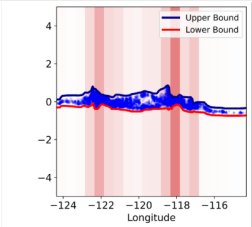
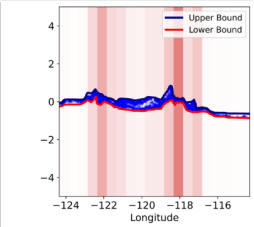
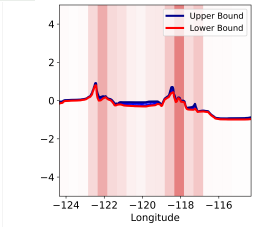
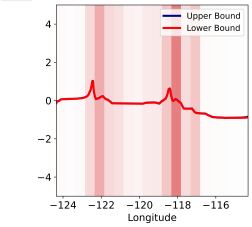
Table 7 shows the performance (RMSE) of NAE-E on the Housing dataset with different variation penalties, along with their corresponding shape plots of the "Longitude" feature. All the tested NAE-E models have a configuration of $K = 32$ and $C = 16$. In the plots, we illustrate the upper and lower bounds for each feature by selecting the maximum and minimum values from the $\frac{K!}{C!(K-C)!}$ possible combinations.

Compared with NAE (Table 2), the performance of NAE-E is slightly worse due to the limitations imposed by its discrete range. However, even without any variation penalty, the estimated bounds by NAE-E still fit the actual score distribution tightly, accurately reflecting the prediction limits.

I PERFORMANCE OF NAE WITH THE SCALING OF EXPERTS

As NAE employs a mixture of experts for data modeling, we explore how the number of total experts (K) and the number of activated experts (C) affect the overall model performance. Table 8 presents the performance of both NAE and NAE-E on the Housing dataset under various configurations. The results indicate that NAE reaches optimal performance at $K = 8$ and $C = 4$, and further increases in K or C do not enhance its performance.

Table 7: Performance (RMSE↓) and explanations of NAE-E with different variation penalties.

$\lambda = \mathbf{0}$	$\lambda = \mathbf{0.1}$	$\lambda = \mathbf{1}$	$\lambda = \mathbf{10}$	$\lambda = \mathbf{100}$
RMSE (↓)				
0.462 ± 0.002	0.460 ± 0.002	0.483 ± 0.001	0.551 ± 0.002	0.585 ± 0.003
Shape Plot				
				

Such a finding is consistent with the role of experts in NAE, which primarily explore the variability in the prediction space to learn the upper and lower bounds, as the dynamic routing mechanism offers a continuous range within the bounds for prediction. In contrast, NAE-E benefits from an increased number of K and C, showing improved performance as these parameters grow. This difference shows how the continuous expert activation in NAE contrasts with the discrete one in NAE-E, highlighting the flexibility of our NAE framework to adapt to various scenarios through different implementations.

Table 8: Results for different numbers of activated experts (C) and all experts (K). The top three scores of each model are highlighted.

$C \setminus K$	**NAE**			
	2	**4**	**8**	**16**
2	0.479 ± 0.003	0.459 ± 0.003	0.455 ± 0.004	0.460 ± 0.004
4	–	0.451 ± 0.002	0.447 ± 0.004	0.455 ± 0.004
8	–	–	0.449 ± 0.004	0.456 ± 0.003
16	–	–	–	0.456 ± 0.005
$C \setminus K$	**NAE-E**			
	16	**32**	**64**	**128**
2	0.486 ± 0.007	0.487 ± 0.005	0.486 ± 0.006	0.489 ± 0.005
4	0.469 ± 0.007	0.466 ± 0.005	0.468 ± 0.004	0.470 ± 0.004
8	0.470 ± 0.003	0.459 ± 0.004	0.458 ± 0.002	0.458 ± 0.004
16	0.587 ± 0.003	0.462 ± 0.002	0.451 ± 0.002	0.453 ± 0.002

To examine the interplay between the number of experts and the variation penalty, we conduct a grid search on the Housing dataset and report the RMSE scores in Table 9. The results indicate that increasing K does not always lead to improved performance, and higher values of K require larger values of λ for regularization to prevent overfitting. This suggests that while more experts can capture more complex patterns, they also increase the risk of overfitting, necessitating stronger regularization to maintain generalization performance.

Table 9: Effect of λ for different values of K.

$K \setminus \lambda$	0	0.1	1	10
2	0.4805	0.4831	0.4989	0.5771
4	0.4498	0.4498	0.4577	0.5068
8	0.4507	0.4470	0.4460	0.4765
16	0.4662	0.4528	0.4464	0.4509

J DISCUSSION ON MODEL COMPLEXITY

As NAE is a fundamental framework aiming to extend generalized additive models, our design prioritizes the performance and comprehensiveness of the overall framework over the optimization of model efficiency. On the basis of Neural Additive Models (NAMs), the neural-network-based Generalized Additive Models (GAMs), NAE introduces more parameters in the expert encoding and dynamic routing steps. To rigorously analyze the additional cost incurred by NAE compared to NAM, suppose we have n features, d-dimensional output for feature encoding, and K total experts for each feature. In our experiments, NAE is implemented with $d = 128$ and $K = 4$. For NAE-D discussed in Appendix G, we have $d = 128$ with $K = 64$. The theoretical and empirical additional costs brought by NAE and its variant are presented in Table 10, including the increase in parameter count and the additional memory required (assuming float32 storage).

Table 10: Theoretical and empirical additional costs brought by NAE compared to NAM. NAE is implemented with $d = 128, K = 4$. NAE-D is implemented with $d = 128, K = 64$. n denotes the number of input features.

	n	**NAE**		**NAE-D**	
		Parameter Count	**Memory Usage**	**Parameter Count**	**Memory Usage**
Housing	8	37k	144k	132k	516k
MIMIC-II	17	157k	613k	281k	1.1M
MIMIC-III	57	1.7M	6.5M	941k	3.59M
Income	14	108k	420k	231k	903k
Credit	30	476k	1.8M	495k	1.89M
Year	90	4.2M	16.0M	1.5M	5.67M
Theoretical	–	$nK[(n+1)d+2]$		$nK(2d+2)$	

The results indicate that the additional cost associated with NAE-D scales linearly with the number of features, whereas the cost for NAE includes a squared term due to its more complex routing mechanism. Specifically, NAE utilizes a full $n \times n$ block matrix to encode feature interactions, which is simplified as a block-diagonal matrix in NAE-D. Despite this increased complexity, the additional memory usage remains manageable with current computing resources. Future efforts could be made to sparsify the routing matrix in NAE, which could potentially reduce the additional cost while retaining the model performance.

Additionally, we conducted experiments to measure actual training time (in seconds) / memory usage (in MB) across varying numbers of features (n) and experts (K) for NAE. Table 11 summarizes the results with 5000 training samples and 100 training epochs, using the encoder architecture from Table 4 with four layers, each containing 128 neurons.

Table 11: Training time / memory usage of NAE with different K and n values.

	$n = 32$	$n = 64$	$n = 128$	$n = 256$	$n = 512$
$K = 4$	39.7s / 2.6MB	39.8s / 7.2MB	42.8s / 22.3MB	64.4s / 76.6MB	174.0s / 281.2MB
$K = 8$	40.4s / 3.6MB	42.6s / 11.2MB	47.6s / 38.4MB	102.3s / 140.8MB	238.0s / 537.7MB
$K = 16$	40.8s / 5.7MB	42.3s / 19.3MB	82.9s / 70.7MB	150.1s / 269.4MB	346.7s / 1050.7MB
$K = 32$	42.9s / 9.8MB	89.6s / 35.6MB	111.7s / 135.2MB	277.5s / 526.4MB	559.0s / 2076.8MB

The results validate the theoretical analysis in Table 10, showing that NAE's training time and memory usage increase with both the number of features and experts. For comparison, we also measured end-to-end training times and memory usage for NA^{2}M, EB^{2}M, NAM, and EBM, comparing them with NAE when $K = 4$, which is the setting used in our main experiments. As the results in Table 12 show, while NAE preserves the capability to model interactions, it presents a computational cost that is lower than NA^{2}M and EB^{2}M, especially when the number of features is large.

Table 12: Training time and memory usage of NA^{2}M, EB^{2}M, NAM, and EBM for varying numbers of features (n). "–" indicates experiments that exceeded one hour to complete.

	$n = 32$	$n = 64$	$n = 128$	$n = 256$	$n = 512$
NA^{2}M	70.9s / 24.2MB	290.7s / 99.0MB	1146.1s / 402.9MB	–	–
EB^{2}M	4.5s / 2.2MB	10.5s / 6.6MB	34.4s / 22.5MB	155.4s / 82.2MB	1004.3s / 313.2MB
NAE ($K = 4$)	39.7s / 2.6MB	39.8s / 7.2MB	42.8s / 22.3MB	64.4s / 76.6MB	174.0s / 281.2MB
NAM	6.3s / 1.6MB	10.4s / 3.1MB	17.2s / 6.3MB	33.0s / 12.9MB	66.0s / 26.8MB
EBM	5.7s / 1.1MB	9.4s / 2.1MB	18.0s / 3.7MB	37.6s / 4.5MB	95.8s / 5.3MB

K MORE VISUALIZATION RESULTS OF NAE ON REAL-WORLD DATA

Figures 11 - 14 show the complete visualization results of feature explanations by NAE on different real-world datasets. Here we present results for datasets with no more than 50 features, including Housing, MIMIC-II, Income, and Credit.

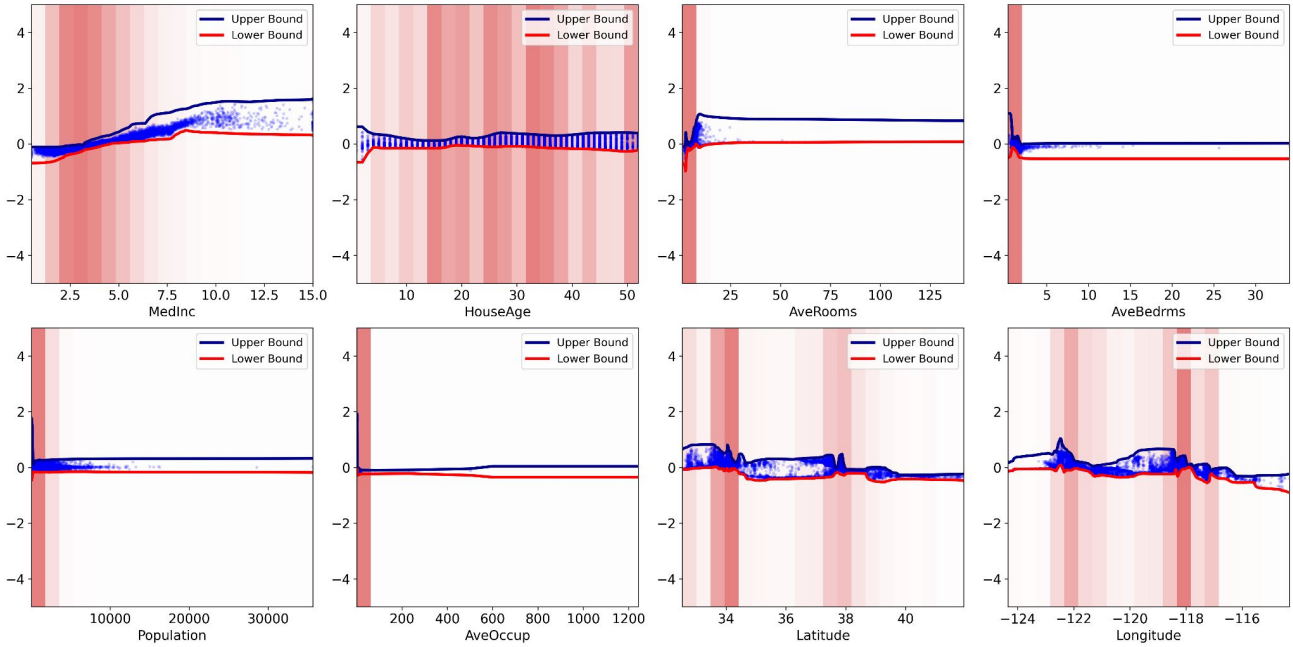


Figure 11: Visualization of feature explanations by NAE on the Housing dataset.

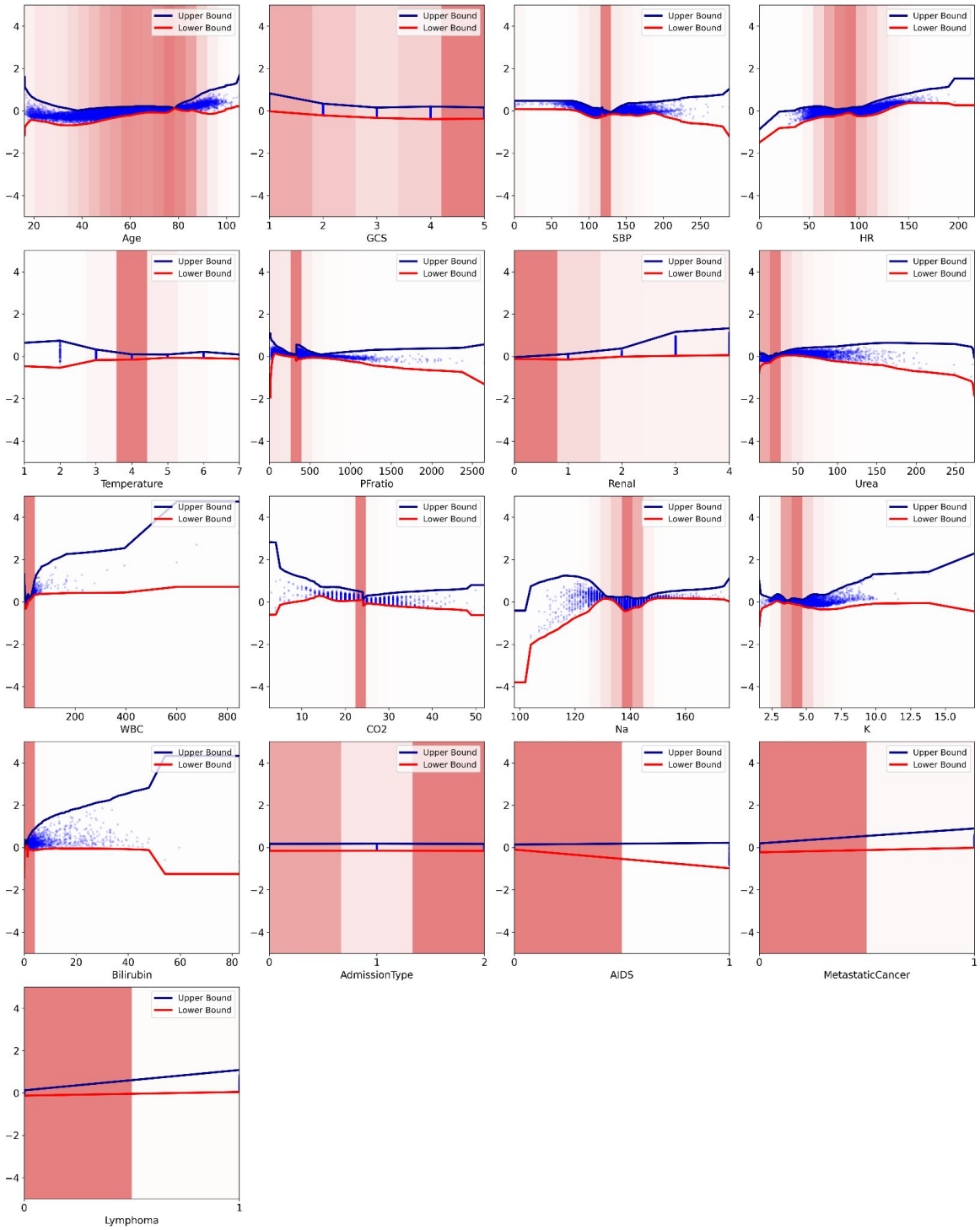


Figure 12: Visualization of feature explanations by NAE on the MIMIC-II dataset.

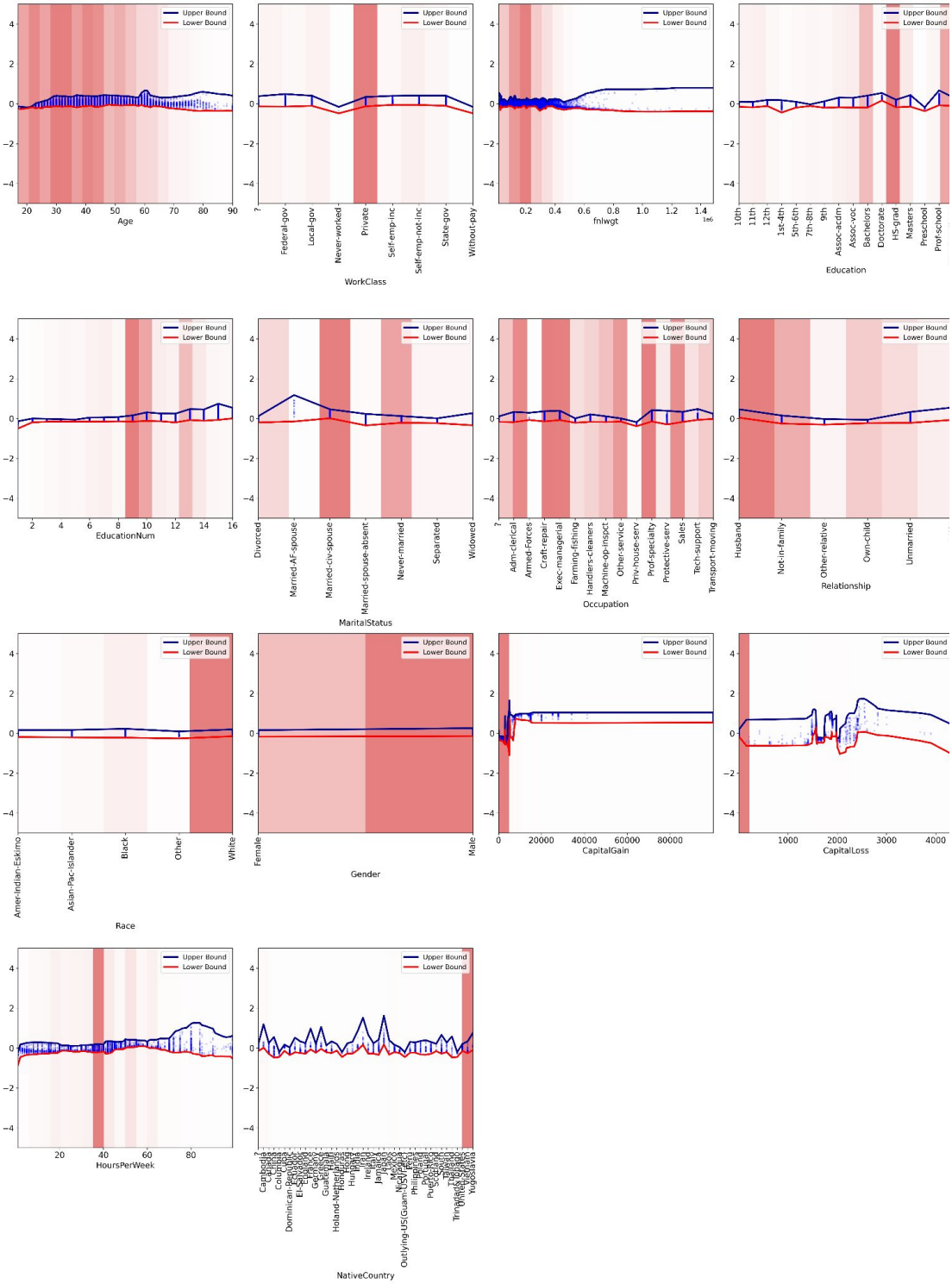


Figure 13: Visualization of feature explanations by NAE on the Income dataset.

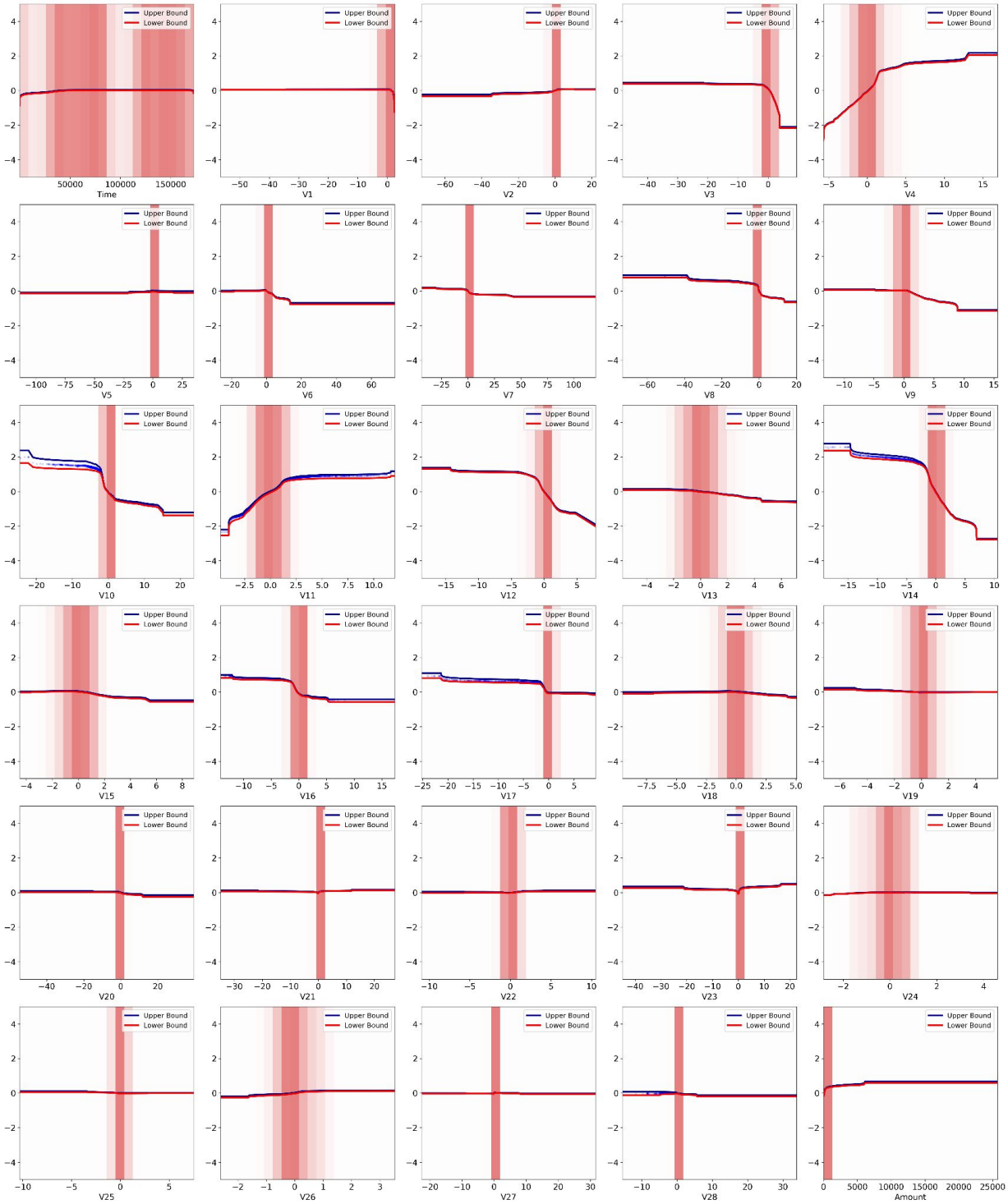


Figure 14: Visualization of feature explanations by NAE on the Credit dataset.