NODE-GAMLSS: Interpretable Uncertainty Modelling via Deep Distributional Regression

Ananyapam De
Institute for Mathematics
TU Clausthal
ananyapam.de@tu-clausthal.de

Anton Thielmann
Institute for Mathematics
TU Clausthal
anton.thielmann@tu-clausthal.de

Benjamin Säfken
Institute for Mathematics
TU Clausthal
benjamin.safken@tu-clausthal.de

Abstract

We propose NODE-GAMLSS, a framework for scalable uncertainty modelling through deep distributional regression. NODE-GAMLSS is an interpretable attention based deep learning architecture which models the location, scale, and shape (LSS) dependent on the data instead of only the conditional mean enabling us to predict quantiles and interpret the feature effects. We perform a benchmark comparison based on simulated and real datasets with state-of-the-art interpretable distributional regression models, demonstrating the superior quantile estimation, accuracy and interpretability. The code is available at https://github.com/AnFreTh/NodeGAMLSS

1 Introduction

Regression analysis traditionally focuses on estimating the conditional mean of a response variable given the explanatory variables. Generalized Additive Models (GAMs) Hastie and Tibshirani are popular interpretable additive mean regression models using smooth functions of covariates. Recently, Agarwal et al. introduced Neural Additive Models (NAMs), which enhance predictive accuracy by utilizing a feedforward neural network for each feature, thereby maintaining interpretability. Chang et al. developed Neural Oblivious Decision Ensembles (NODE-GAM), which adapt the NODE architecture by Popov et al. to a GAM, maintaining predictive accuracy while preserving interpretability.

While these are powerful mean prediction models, they do not capture the full conditional distribution of the response. This is problematic when quantifying uncertainty is crucial, such as with heteroskedasticity or heavy-tailed distributions. The forumlation of Generalized Additive Models for Location, Scale, and Shape (GAMLSS) Rigby and Stasinopoulos [2005] introduced distributional regression which extend GAMs to model other parameters of the response, such as variance or skewness, as functions of the explanatory variables. mBoostLSS [Hofner et al., 2016] offers alternatives with shrinkage and variable selection.

Other methods for distributional modelling include conditional transformation models Hothorn et al. [2013], quantile Koenker [2005], and expectile regression [see Newey and Powell, 1987, Kneib et al., 2023]. Recent advancements present deep distributional models such as XGBoostLSS März [2019] and LightGBMLSS März [2023], which extend boosting algorithms Chen and Guestrin [2016], Ke

Workshop on Bayesian Decision-making and Uncertainty, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

et al. [2017] into a probabilistic framework, though at the cost of interpretability. NAMLSS Thielmann et al. [2024] provides feature level interpretability and leverages flexible scalable feature functions like MLPs and Transformers within the NAM framework. NODE-GAMLSS integrates the NODE-GAM architecture for distributional regression by predicting the conditional distributional parameters. These parameters are then optimized with a loss function, thereby offering an interpretable and scalable framework for capturing the distributional properties of the response variable.

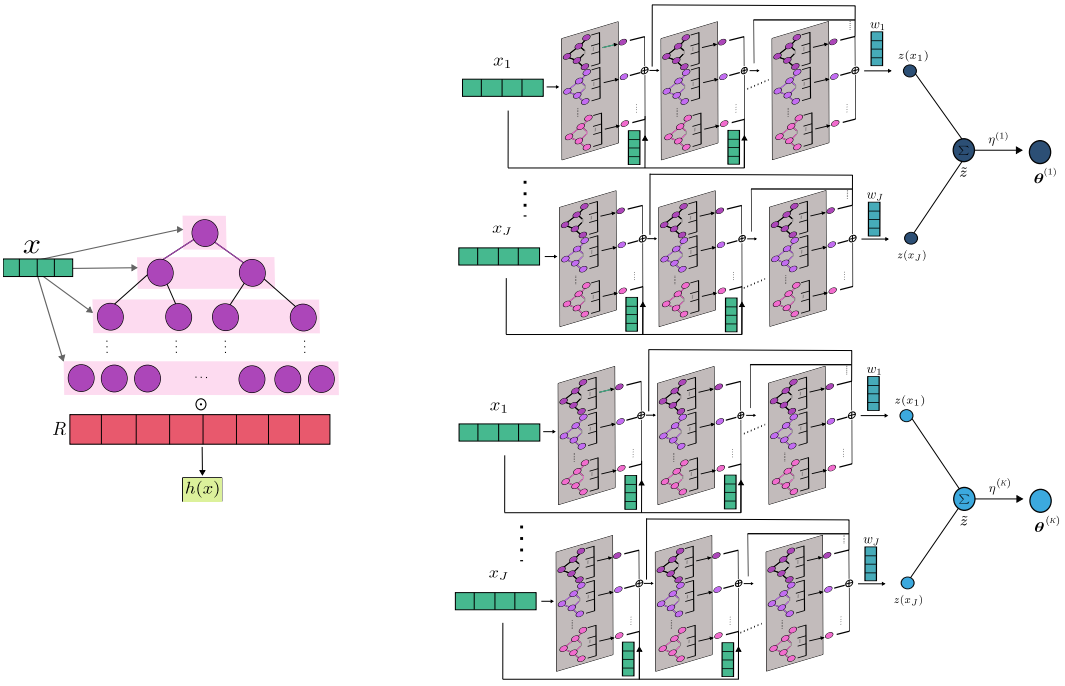

Figure 1: An oblivious decision tree (left) with all nodes at the same depth using identical features and thresholds with the input feature being passed at every depth. For each feature, multiple trees form a layer that are stacked together, with the respective input feature passed through every layer (right). The output of an ensemble is the sum of the weighted average of all the trees for a feature, which are then applied the respective activation. The additivity constraint here prevents overfitting.

2 Methodology

Suppose we are given covariates $\mathbf{X} = \{(x_{1i}, x_{2i}, \cdots, x_{Ji})\}_{i=1}^{n}$ representing J input features $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_J$ with each $\mathbf{x}_j \in \mathbb{R}^n$ and the target $\mathbf{y} = \{y_i\}_{i=1}^{n}$. We assume $\mathbf{y} \sim \mathcal{D}(\boldsymbol{\theta}(\mathbf{X}))$, where $\boldsymbol{\theta}(\mathbf{X}) = (\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(K)})$ are K parameters of the response distribution $\mathcal{D}$. We model the dependence of the k distributional parameters as

$$\theta^{(k)} = \eta^{(k)}(\beta^{(k)} + \sum_{j=1}^{J} z_j^{(k)}(x_j))$$

where $\beta^{(k)}$ is the intercept, z_j are feature functions, $\eta^{(k)}$ are activation functions. The function z_j consists of L layers, each with I differentiable Oblivious Decision Trees (ODTs) of depth c handling both real and vector-valued inputs, enabling scalable training by processing data in batches. All nodes in a tree share the same feature function, and nodes at the same depth use identical input features and thresholds (Figure 1). Each ODT in layer l of depth c compares c selected features of an input $\boldsymbol{x} \in \mathbb{R}^d$ against thresholds b^c, using feature function $F : \mathbb{R}^d \to \mathbb{R}$ where $d = n + l - 1$. The output $h(\boldsymbol{x})$ is the inner product $\odot$ of a response vector $R \in \mathbb{R}^{2^c}$ with the results of these comparisons:

$$h(\boldsymbol{x}) = R \odot \left(\bigotimes_{c=1}^{C} \begin{bmatrix} \sigma\left(F(\boldsymbol{x}) \leq b^c\right) \\ \sigma\left(F(\boldsymbol{x}) > b^c\right) \end{bmatrix} \right),$$

where σ is the entmoid, $\otimes$ is outer product. For tree i in layer l, the feature function $F_{li}(\boldsymbol{x})$ is:

$$F_{li}(\boldsymbol{x}) = \sum_{j=1}^{d} x_j \cdot G_{ij} + \frac{1}{\sum_{\hat{l}=1}^{l-1}\sum_{\hat{i}=1}^{I} g_{l\hat{l}i}} \sum_{\hat{l}=1}^{l-1}\sum_{\hat{i}=1}^{I} h_{\hat{l}\hat{i}}(\boldsymbol{x}) g_{l\hat{l}i} a_{l\hat{l}i}$$

Here, $G_i = \text{entmax}_\alpha(F_i/T)$, $g_{li\hat{i}} = G_i G_{\hat{i}}$, and $a_{l\hat{l}i}$ are attention weights that focus on specific trees. The temperature T anneals to zero, forcing G_i to become one-hot, making $g_{li\hat{i}} = 1$ only when $G_i = G_{\hat{i}}$ thus acting as an additive model. Outputs from previous layers become inputs to the next. For an input $\boldsymbol{x}$, the layer inputs $\boldsymbol{x}^l$ are

$$\boldsymbol{x}^1 = \boldsymbol{x}, \quad \boldsymbol{x}^l = \left[\boldsymbol{x}, h^1(\boldsymbol{x}^1), \ldots, h^{(l-1)}(\boldsymbol{x}^{(l-1)})\right]$$

for $l > 1$. The final prediction is given by

$$z^{(k)}(\boldsymbol{x}) = \frac{1}{LI}\sum_{l=1}^{L}\sum_{i=1}^{I} h_i^l(\boldsymbol{x}^l) w_{li}.$$

To ensure the constraints of $\mathcal{D}$, the model is adapted to a Location, Scale and Shape (LSS) framework, where the output is passed through the activation $\eta^{(k)}$, which transforms the parameters. We minimize the sum of the negative log likelihood of the predicted distribution with respect to the observed values: $l(\boldsymbol{\theta}) = \sum_{i=1}^{n} -\log(\mathcal{L}(\boldsymbol{\theta}|y_i))$. Since h is differentiable, the model is trained end-to-end using backpropagation and gradient descent Kingma and Ba [2017].

3 Experiments

Feature interpretability and interactions: NODE-GAMLSS offers feature-level interpretability enabling visual analysis illustrated for the California housing dataset in Figure 2.

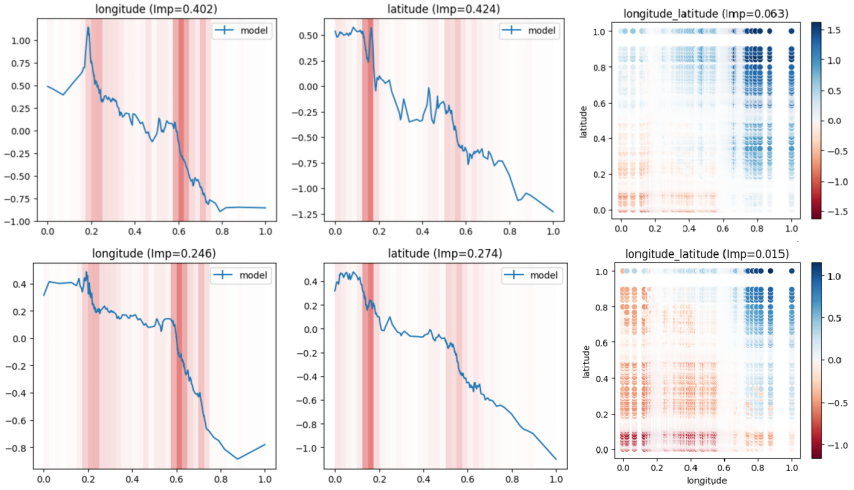

Figure 2: The rows display raw mean and variance predictions respectively. The left plots illustrate single feature effects for longitude, latitude and median age, with pink bars indicating normalized data density. The right plots highlight the interaction effects of longitude and latitude. The model excels in capturing jagged functions, evident in the sharp price jumps around San Francisco and Los Angeles. The second row shows decreasing variance in areas further from large cities.

Real world data: We compare our model against state-of-the-art interpretable distributional models—NAMLSS, GAMBoostLSS, and GAMLSS. NODE-GAMLSS outperforms benchmarks (Pace and Barry [1997], Nash et al. [1995], Lantz [2013]) achieving the lowest NLL, MSE and MAE (in Table 1) indicating better model fit and predictive accuracy.

Table 1: Comparison on real datasets using a normal response

Dataset	Model	NLL	MSE	MAE
Abalone	NODE-GAMLSS	**0.951 ± 0.024**	**0.492 ± 0.044**	**0.494 ± 0.022**
	NAMLSS	1.078 ± 0.079	1.076 ± 0.098	0.787 ± 0.036
	GAMLSS	1.056 ± 0.034	0.483 ± 0.032	0.501 ± 0.017
	GAMBoostLSS	0.998 ± 0.031	0.536 ± 0.046	0.732 ± 0.032
California	NODE-GAMLSS	**0.726 ± 0.023**	**0.314 ± 0.011**	**0.392 ± 0.004**
	NAMLSS	0.785 ± 0.047	0.758 ± 0.051	0.654 ± 0.026
	GAMLSS	0.917 ± 0.020	0.366 ± 0.015	0.442 ± 0.004
	GAMBoostLSS	1.025 ± 0.182	0.420 ± 0.011	0.648 ± 0.009
Insurance	NODE-GAMLSS	**0.556 ± 0.147**	**0.151 ± 0.027**	**0.226 ± 0.024**
	NAMLSS	0.653 ± 0.057	0.655 ± 0.069	0.568 ± 0.040
	GAMLSS	0.732 ± 0.048	0.253 ± 0.024	0.503 ± 0.024
	GAMBoostLSS	0.644 ± 0.068	0.269 ± 0.028	0.518 ± 0.028

Table 2: Comparison of quantile estimation metrics

Model	CRPS	Quantile Score	Coverage Probability
NODE-GAMLSS	**0.2536 ± 0.0101**	**0.1301 ± 0.0013**	**0.956 ± 0.013**
NAMLSS	0.3224 ± 0.0202	0.1654 ± 0.0028	0.932 ± 0.012
GAMLSS	0.2795 ± 0.0132	0.1433 ± 0.0032	0.914 ± 0.025
GAMBoostLSS	0.3469 ± 0.0154	0.1780 ± 0.0082	0.895 ± 0.053

Quantile estimation: Our model captures the full conditional distribution, enabling quantile estimation illustrated for the California housing dataset as shown in Table 3 achieving the lowest quantile score and CRPS, and the highest coverage. Figure 3 compares performance across quantiles.

Feature learning: We simulate $x_1, x_2 \sim \mathcal{U}(-5, 5)$ and define $f_1(x) = \sin(x)$, $f_2(x) = 2x$, $f_3(x) = x^2$, $f_4(x) = e^x$ such that $y \sim \mathcal{N}(f_1(x_1) + f_2(x_2), f_3(x_1) + f_4(x_2))$. The effects learnt by NODE-GAMLSS shown in Figure 4 clearly capture the shapes of the original functions.

Distribution modelling: We synthetically generate $n = 3000$ observations with $J = 5$ features from Normal, Poisson, Lognormal, and Gamma distributions (see Appendix A.5). NODE-GAMLSS achieves state-of-the-art performance, matching NAMLSS on both count and continuous data.

Table 3: Comparison of negative log-likelihood for multiple distributions

Model	Gamma	Normal	Poisson	Lognormal
NODE-GAMLSS	**0.833 ± 0.072**	**1.422 ± 0.027**	**1.375 ± 0.024**	**1.514 ± 0.033**
NAMLSS	0.827 ± 0.076	1.442 ± 0.048	1.386 ± 0.035	1.504 ± 0.033
GAMLSS	1.064 ± 0.033	1.432 ± 0.039	1.383 ± 0.005	1.535 ± 0.068
GAMBoostLSS	3.132 ± 0.284	1.491 ± 0.011	1.379 ± 0.045	1.524 ± 0.021

Acknowledgments and Disclosure of Funding

Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within project 450330162 is gratefully acknowledged.

4 Limitations

A crucial aspect in applying our proposed method, as well as other distributional methods, is selecting the appropriate distributional assumptions. Hence this approach necessitates some understanding and domain knowledge of the data distribution. Also rather than minimizing an error measure, our method primarily uses the negative log likelihood, a strictly proper scoring rule Lakshminarayanan

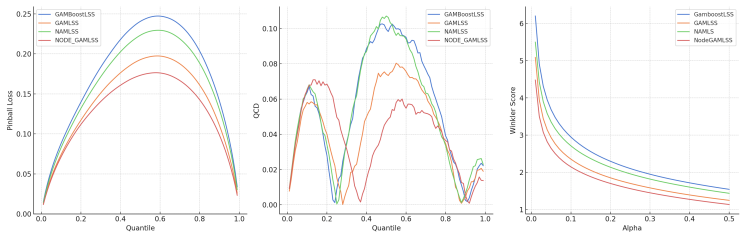

Figure 3: NODE-GAMLSS outperforms other models, with the lowest pinball loss (left), minimal QCD for most quantiles (middle), and the lowest Winkler scores (right).

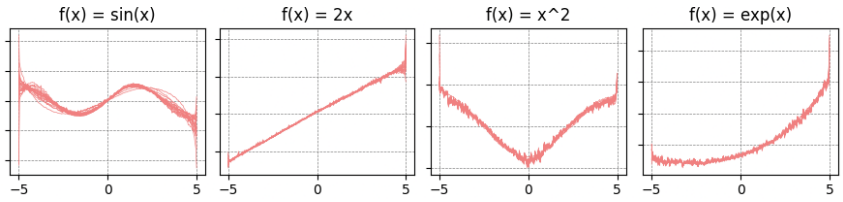

Figure 4: Feature effects learnt for 20 individual runs.

et al. [2017], minimized in expectation if and only if the conditional density matches the underlying data distribution Hastie et al. [2001].

5 Conclusion and Future Work

NODE-GAMLSS presents a significant advancement in uncertainty modelling via distributional regression, providing interpretability of covariate effects and exceptional predictive accuracy. These qualities make NODE-GAMLSS highly suitable for applications in high-risk domains, where understanding and mitigating uncertainty is crucial. Using other types of flexible distributions like mixture density networks Seifert et al. [2022] or normalizing flows Papamakarios et al. [2021] are apparent extensions. Multivariate responses conditionally dependent on covariates can be modeled using a copula-based approach for NODE-GAMLSS that would significantly improve the general applicability. While initially designed for tabular data, NODE-GAMLSS can be naturally extended to handle multimodal data by integrating components such as a CNN or Transformers as feature networks for image and text input respectively.

References

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. NODE-GAM: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=g8NJR6fCCl8`.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL `http://dx.doi.org/10.1145/2939672.2939785`.

Hadi Fanaee-T. Bike Sharing. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL `https://doi.org/10.1198/016214506000001437`.

PJ Green and TJ Cole. Smoothing reference centile curves: the lms method and penalized likelihood. *Statistics in Medicine*, 11:1305 – 1319, 1992. ISSN 1097-0258.

Trevor Hastie and Robert Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3):297 – 310, 1986. doi: 10.1214/ss/1177013604. URL https://doi.org/10.1214/ss/1177013604.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Benjamin Hofner, Andreas Mayr, and Matthias Schmid. gamboostlss: An r package for model building and variable selection in the gamlss framework. *Journal of Statistical Software*, 74(1):1–31, 2016. doi: 10.18637/jss.v074.i01. URL https://www.jstatsoft.org/index.php/jss/article/view/v074i01.

Torsten Hothorn, Thomas Kneib, and Peter Bühlmann. Conditional transformation models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):3–27, March 2013. ISSN 1467-9868. doi: 10.1111/rssb.12017. URL http://dx.doi.org/10.1111/rssb.12017.

Guolin Ke, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIP 2017)*, December 2017. URL https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Thomas Kneib, Alexander Silbersdorff, and Benjamin Säfken. Rage against the mean – a review of distributional regression approaches. *Econometrics and Statistics*, 26:99–123, 2023. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2021.07.006. URL https://www.sciencedirect.com/science/article/pii/S2452306221000824.

Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

Roger W Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:46:y:1978:i:1:p:33-50.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Brett Lantz. *Machine Learning with R*. Packt Publishing, 2013. ISBN 1782162143.

Simon Kristoffersson Lind, Ziliang Xiong, Per-Erik Forssén, and Volker Krüger. Uncertainty quantification metrics for deep regression, 2024. URL https://arxiv.org/abs/2405.04278.

Alexander März. LightGBMLSS: An Extension of LightGBM to Probabilistic Modelling. https://github.com/StatMixedML/LightGBMLSS, 2023. GitHub repository, Version 0.4.0.

Alexander März. Xgboostlss – an extension of xgboost to probabilistic forecasting, 2019. URL https://arxiv.org/abs/1907.03178.

Warwick Nash, Tracy Sellers, Simon Talbot, Andrew Cawthorn, and Wes Ford. Abalone. UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C55C7W.

Whitney K. Newey and James L. Powell. Asymmetric least squares estimation and testing. *Econometrica*, 55(4):819–847, 1987. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911031.

Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics and Probability Letters*, 33(3):291–297, 1997. URL https://EconPapers.repec.org/RePEc:eee:stapro:v:33:y:1997:i:3:p:291-297.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021. URL https://arxiv.org/abs/1912.02762.

Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data, 2019. URL https://arxiv.org/abs/1909.06312.

R. A. Rigby and D. M. Stasinopoulos. Generalized Additive Models for Location, Scale and Shape. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 54(3):507–554, 04 2005. ISSN 0035-9254. doi: 10.1111/j.1467-9876.2005.00510.x. URL https://doi.org/10.1111/j.1467-9876.2005.00510.x.

Quentin Edward Seifert, Anton Thielmann, Elisabeth Bergherr, Benjamin Säfken, Jakob Zierk, Manfred Rauh, and Tobias Hepp. Penalized regression splines in mixture density networks, 2022. URL https://doi.org/10.21203/rs.3.rs-2398185/v1.

Anton Frederik Thielmann, René-Marcel Kruse, Thomas Kneib, and Benjamin Säfken. 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, 2024.

Rainer Winkelmann and Stefan Boes. Analysis of microdata. *Analysis of Microdata*, 01 2006. doi: 10.1007/3-540-29607-7.

Robert L. Winkler. A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972. doi: 10.1080/01621459.1972.10481224. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1972.10481224.

A Benchmark Details

We use the same data-preprocessing, model specifications and initializations for all the experiments. For the experiments on real world data, quantile estimation and distribution modelling above, a 5-fold cross-validation is performed, with results reported as the mean and standard deviations across all metrics and datasets.

A.1 Data preprocessing

All numerical variables are scaled between 0 and 1 using `MinMaxScaler`, and all categorical features are one-hot encoded. We implement minor quantile smoothing (`quantile_noise=0.001`), leveraging one of the key strengths of NODE-GAMLSS in modelling jagged shape functions. For implementing distributions not having the entire real line as the support, it is essential to maintain the original support. We do not transform responses in such cases (e.g. Lognormal, Poisson or Gamma).

A.2 Model specifications

The NODE-GAMLSS model is configured with the following parameters:

- **Model Structure**: The model employs a total of 75 trees (`num_trees=75`, denoted as I), each with a depth of 3 (`tree_depth=3`, denoted as C). It comprises 2 layers (`num_layers=2`, denoted as L) without additional tree dimensions (`addi_tree_dim=0`). The feature sampling per tree is set to 50% (`colsample_bytree=0.5`). The attention embedding dimension is set to 8 (`dim_att=8`).
- **Optimization and Regularization**: We use the Adam optimizer where the learning rate is initialized at 0.001 (`lr=0.001`) with a batch size of 2048 (`batch_size=2048`). Although L^2 regularization is omitted `l2_lambda=0`, dropout is applied with a rate of 0.3 on the last layer (`last_dropout=0.3`), while no output dropout is used (`output_dropout=0`).
- **Learning Rate Scheduling**: The optimization process includes a learning rate decay mechanism, where the learning rate (`lr=0.01`) is warmed up over 100 steps (`lr_warmup_steps=100`) and decays every 300 steps (`lr_decay_steps=300`). Additionally, the learning rate is annealed over 2000 steps (`anneal_steps=2000`).
- **Training Procedure**: Early stopping is implemented after 2000 steps without improvement (`early_stopping_steps=2000`). The training will continue for a maximum of 20,000 steps (`max_steps=20000`) or up to 20 hours (`max_time=20 \times 3600`), whichever occurs first. The last 5 checkpoints are saved (`n_last_checkpoints=5`) to ensure stability.
- **Reproducibility**: To ensure reproducibility, a random seed 1377 is used (`seed=1377`).

A.3 Initializations

The initialization of selection logits, thresholds, response values, and temperature parameters is handled as follows:

- **Selection Logits**: Each feature selection logit $F \sim \mathcal{U}(0, 1)$ is initialized with a uniform.
- **Response**: The response values $R \sim \mathcal{N}(0, 1)$ are initialized as a Gaussian.
- **Threshold initialization parameter**: The threshold initialization parameter β determines the distribution of the initial thresholds. If $\beta = 1$, the initial thresholds will have the same distribution as the data points. If $\beta > 1$ (e.g., 10), the thresholds will be closer to the median of the data values. If $\beta < 1$ (e.g., 0.1), the thresholds will approach the minimum or maximum of the data values.
- **Thresholds**: The thresholds b are initialized using random percentiles of the data, calculated as $b \sim 100 \times \mathcal{B}(\beta, \beta)$, where $\mathcal{B}$ denotes the Beta distribution.
- **Threshold cutoff temperature**: The cutoff temperature, denoted by $\tau \in \mathbb{R}^+$, is set to 1 by default. This parameter is used to scale the temperatures within the model.
- **Temperature**: The logarithmic temperatures are initialized to ensure that all bin selectors fall within the linear region of the sparse-sigmoid function. This is done by computing a

specific percentile of the absolute differences between feature values and their corresponding thresholds. The chosen percentile is given by $\min(\tau, 1)$ and the scaling factor is $\max(\tau, 1)$.

  - $\tau > 1.0$: A margin is created between data points and the sparse-sigmoid cutoff value, ensuring that all points are mapped within $\left(0.5 - \frac{0.5}{\tau}\right)$ and $\left(0.5 + \frac{0.5}{\tau}\right)$.
  - $\tau < 1.0$: A portion of the data points corresponding to $(1 - \tau)$ will fall into the flat region of the sparse-sigmoid function. For example, if $\tau = 0.9$, 10% of the points will be mapped to either 0.0 or 1.0.
  - $\tau = 1.0$: This represents a balanced scenario where the data points are neither compressed nor expanded relative to the sparse-sigmoid cutoff.
- **Attention**: The query and the key for the attention blocks are set to 0 and the selection logits are sampled with a uniform $\mathcal{U}(0, 1)$.
- **Weights**: The weights w are Glorot initialized using a Xavier Uniform distribution with the default gain set to 1.

A.4 Competing model training specifications

To ensure a fair and consistent comparison across different models, we align the configurations of our model with those of NAMLSS, GAMLSS, and GAMBoostLSS wherever possible. Each model was configured for the respective response distribution when comparing. Both the neural models NODE-GAMLSS and NAMLSS were configured with identical learning rates, batch sizes, optimizers and activation functions. For NAMLSS, we adopted the hyperparameters as specified in Table 9 of Thielmann et al. [2024], using the first proposed architecture and all feature functions were modeled as Multi-Layer Perceptrons (MLP). For the GAMLSS model, we used the default hyperparameters provided in the GAMLSS R package. In cases where the RS solver failed to converge, CG solver Green and Cole [1992] was used. In the case of GAMBoostLSS, the primary hyperparameter we configured was `mstops=500` but the model often encountered difficulties due to rank deficiency in the data. GAMBoostLSS includes boosting approaches based on GAMs and GLMs and we selected the boosting method yielding the highest log-likelihood.

A.5 Simulated data generation

We consider $J = 5$ features with $n = 3000$ observations across synthetic datasets generated from Normal, Poisson, Lognormal, and Gamma distributions, to demonstrate the performance on both count and continuous data. The features x_1, x_2, x_3, x_4, and x_5 are independently sampled from a uniform distribution $U(0, 1)$. Two parameters θ_1 and θ_2 are computed as:

$$\theta_1 = \frac{21}{20}|x_3| + 3\cos\left(\frac{3}{2}x_2\right) - 2x_5 - 0.2e^{x_1} - x_4^2,$$

$$\theta_2 = \exp\left(-0.004x_4 + (x_1 - 0.2)^2 - 1.5x_2\right) + 0.0005x_5 - 0.2x_3,$$

followed by the transformations $\theta_{1,\text{positive}} = \log(1 + e^{\theta_1})$ and $\theta_{2,\text{positive}} = \log(1 + e^{\theta_2})$ which are used to generate response variables y based on different distributions:

- **Poisson**: $y \sim \text{Poisson}(\theta_{1,\text{positive}})$.
- **Normal**: $y \sim \text{Normal}(\theta_1, \sqrt{\theta_{2,\text{positive}}})$.
- **Gamma**: $y \sim \text{Gamma}(\theta_{1,\text{positive}}, \theta_{2,\text{positive}})$.
- **Lognormal**: $y \sim \text{Lognormal}(\log(\theta_{1,\text{positive}}), \sqrt{\theta_{2,\text{positive}}})$.

B Objective measures

We detail the objective measures used for evaluating the performance of our model, focusing on deviance metrics, negative log-likelihoods for the distributions, and parameter constraints enforced through activation functions.

B.1 Deviance metrics

Beyond metrics like MSE, MAE, we require the following metrics to evaluate quantile estimates.

B.1.1 Quantile Score (QS)

The Quantile Score for a given quantile τ is defined as:

$$\text{QS}_\tau(y, \hat{F}) = \frac{1}{n} \sum_{i=1}^{n} [\tau \cdot (y_i - \hat{q}_{\tau,i}) \cdot \mathbb{I}(y_i \geq \hat{q}_{\tau,i}) + (1 - \tau) \cdot (\hat{q}_{\tau,i} - y_i) \cdot \mathbb{I}(y_i < \hat{q}_{\tau,i})]$$

where y_i is the observed value, $\hat{q}_{\tau,i}$ is the predicted τ-th quantile, and $\mathbb{I}(\cdot)$ is the indicator function.

B.1.2 Continuous Ranked Probability Score (CRPS)

The CRPS by Gneiting and Raftery [2007] is defined as:

$$\text{CRPS}(y, \hat{F}) = \int_{-\infty}^{\infty} \left[\hat{F}(x) - \mathbb{I}(y \leq x)\right]^2 dx$$

where $\hat{F}$ is the cumulative distribution function (CDF) of the predicted distribution, and $\mathbb{I}(y \leq x)$ is the indicator function that equals 1 if $y \leq x$ and 0 otherwise.

B.1.3 Pinball Loss (PL)

The Pinball Loss Koenker and Bassett [1978] for a given quantile τ is defined as:

$$\text{PL}_\tau(y, \hat{q}_\tau) = |\tau \cdot (y - \hat{q}_\tau) \cdot \mathbb{I}(y \geq \hat{q}_\tau) + (1 - \tau) \cdot (\hat{q}_\tau - y) \cdot \mathbb{I}(y < \hat{q}_\tau)|$$

where $\hat{q}_\tau$ is the predicted τ-th quantile.

B.1.4 Winkler Score (WS)

The Winkler Score Winkler [1972] for an interval $[u_i, l_i]$ is given by:

$$W_\alpha = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} (u_i - l_i) + \frac{2}{\alpha}(l_i - y_i), & \text{if } y_i < l_i \\ (u_i - l_i), & \text{if } l_i \leq y_i \leq u_i \\ (u_i - l_i) + \frac{2}{\alpha}(y_i - u_i), & \text{if } y_i > u_i \end{cases}$$

B.2 Log-likelihoods

While the choice of loss can depend on the specific task, negative log-likelihood (NLL) is often preferred for uncertainty modelling [Lind et al., 2024]. Below, we detail the Log-likelihoods for the distributions currently implemented in NODE-GAMLSS.

B.2.1 Normal distribution

$$\log\left(\mathcal{L}\left(\mu, \sigma^2 \mid y\right)\right) = -\frac{n}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \mu\right)^2$$

Here, $\mu \in \mathbb{R}$ is the location parameter, and $\sigma \in \mathbb{R}^+$ is the scale parameter.

B.2.2 Poisson distribution

$$\log(\mathcal{L}(\lambda \mid x)) = \sum_{i=1}^{n} \left[x_i \log(\lambda) - \lambda - \log\left(x_i!\right)\right]$$

Here, x_i are non-negative integers, and $\lambda \in \mathbb{R}^+$ is the rate parameter.

B.2.3 Inverse Gamma distribution

$$\log(\mathcal{L}(\alpha, \beta \mid y)) = -n(\alpha + 1)\overline{\log y} - n \log \Gamma(\alpha) + n\alpha \log \beta - \sum_{i=1}^{n} \beta y_i^{-1}$$

Here, $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter.

B.2.4 Beta distribution

$$\log(\mathcal{L}(\alpha, \beta \mid x)) = \sum_{i=1}^{n} [(\alpha - 1) \log x_i + (\beta - 1) \log(1 - x_i)] - n \log B(\alpha, \beta)$$

Here, $\alpha > 0$ is the first shape parameter and $\beta > 0$ is the second shape parameter.

B.2.5 Dirichlet distribution

$$\log(\mathcal{L}(\alpha \mid x)) = \log \Gamma \left(\sum_{i=1}^{k} \alpha_i \right) - \sum_{i=1}^{k} \log \Gamma(\alpha_i) + \sum_{i=1}^{k} (\alpha_i - 1) \log x_i$$

Here, $\alpha_i > 0$ are the parameters of the distribution, which must be positive, and $x_i \geq 0$ are the observations, which must be non-negative and sum to 1.

B.2.6 Gamma distribution

$$\log(\mathcal{L}(\alpha, \beta \mid x)) = n\alpha \log \beta - n \log \Gamma(\alpha) + (\alpha - 1) \sum_{i=1}^{n} \log x_i - \beta \sum_{i=1}^{n} x_i$$

Here, $\alpha > 0$ is the shape parameter and $\beta > 0$ is the rate parameter.

B.2.7 Lognormal distribution

$$\log(\mathcal{L}(\mu, \sigma \mid x)) = -\frac{n}{2} \log(2\pi) - n \log \sigma - \sum_{i=1}^{n} \log x_i - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (\log x_i - \mu)^2$$

Here, $\mu \in \mathbb{R}$ is the location parameter and $\sigma > 0$ is the scale parameter. The observations $x_i > 0$ are positive.

B.2.8 Student's T distribution

$$\log(\mathcal{L}(\nu, \mu, \sigma \mid x)) = \sum_{i=1}^{n} \log \left[\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\sigma\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(x_i - \mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \right]$$

Here, $\nu > 0$ is the degrees of freedom, $\mu \in \mathbb{R}$ is the location parameter, and $\sigma > 0$ is the scale parameter.

B.2.9 Negative Binomial distribution

$$\log(\mathcal{L}(r, p \mid x)) = \sum_{i=1}^{n} [\log \Gamma(x_i + r) - \log \Gamma(x_i + 1) - \log \Gamma(r) + r \log(1 - p) + x_i \log p]$$

Here, $r > 0$ is the number of successes, $0 < p < 1$ is the probability of success, and x_i are counts.

B.2.10 Categorical distribution

$$\log(\mathcal{L}(\theta \mid x)) = \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \log \theta_j$$

Here, $\theta_j \geq 0$ are the probabilities for each category, which must be non-negative and sum to 1, and x_{ij} is an indicator variable that is 1 if observation i is in category j, and 0 otherwise.

C Activation Functions

The common parameter constraints we come across are positivity which is implemented using the softplus$(x) = \log(1 + e^x)$. For the categorical distribution, the probabilities must be normalized which is done using softmax. For instance, consider the Normal distribution: the location parameter $\theta^{(1)} = \mu \in \mathbb{R}$ is unrestricted, whereas the scale parameter $\theta^{(2)} = \sigma \in \mathbb{R}^+$ must be positive. To enforce this positivity constraint, we use the Softplus activation function for $\theta^{(2)}$. Consequently, our transformed parameter predictions are: $\hat{\mu} = \hat{\theta}^{(1)} = \eta^{(1)}(\tilde{\theta}^{(1)}) = \tilde{\theta}^{(1)}$ and $\hat{\sigma} = \hat{\theta}^{(2)} = \ln(1 + e^{\tilde{\theta}^{(2)}})$

D Illustrative Results

We illustrate some additional use cases of our model visually and present some further benchmarks.

D.1 Further Benchmarks

We perform 5-fold cross-validation and benchmark our model using the Bike Sharing Fanaee-T [2013] and German Socio-Economic Panel 1994-2002 datasets Winkelmann and Boes [2006]. We apply a Poisson distribution for the Bike Sharing dataset and a Lognormal distribution for the German dataset based on the response values.

Table 4: Comparison on real datasets

Dataset	Model	NLL
Bike Sharing	NODE-GAMLSS	25.613 ± 5.739
	NAMLSS	29.176 ± 11.307
	GAMLSS	36.004 ± 9.608
GSOEP9402	NODE-GAMLSS	250.768 ± 25.342
	NAMLSS	282.131 ± 19.421
	GAMLSS	300.182 ± 9.789

As shown in Table 4, NODE-GAMLSS demonstrates substantially better values of the NLL demonstrating a superior ability to learn the shape of the response using the covariates. The GAMLSS model often faces convergence issues with both the RS as well as the CG solver and hence averaging is not done across all five folds. GAMBoostLSS does is not able to execute due to rank deficiency issues.

D.2 Uncertainty modelling and probabilistic forecasts

The main goal of NODE-GAMLS is uncertainty modelling. In Figure 5, we use a ridge plot to visualize the predicted uncertainties on a subset of 15 datapoints from the California housing dataset. We can visually see that NODE-GAMLSS exhibits greater certainty around the true values compared to the other models as evidenced by the low Winkler scores in Table 2. This can also be used to provide a probabilistic forecast, enabling the derivation of various quantities of interest. Figure 6 illustrates a subset of 15 predictions, showcasing the model's ability to estimate different quantiles.

D.3 Feature Importance and Interpretability

The model computes the feature importance as the weighted average of the absolute value of the response weighted by the counts of each unique value in the purified data.

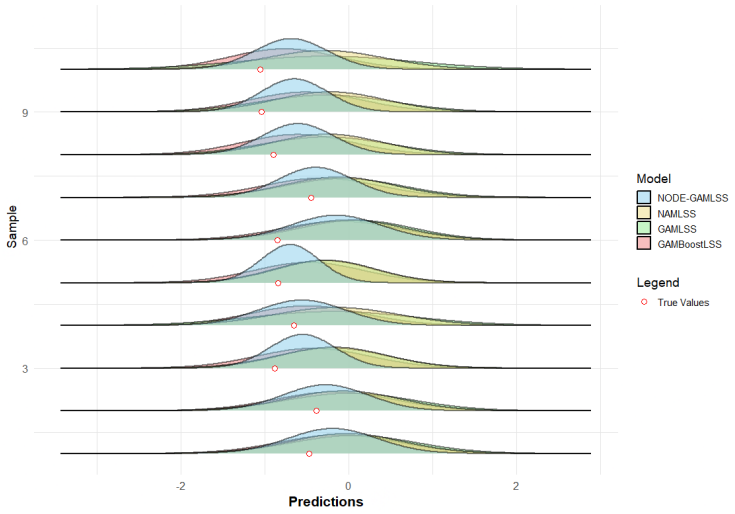

Figure 5: Distributional predictions for California housing dataset.

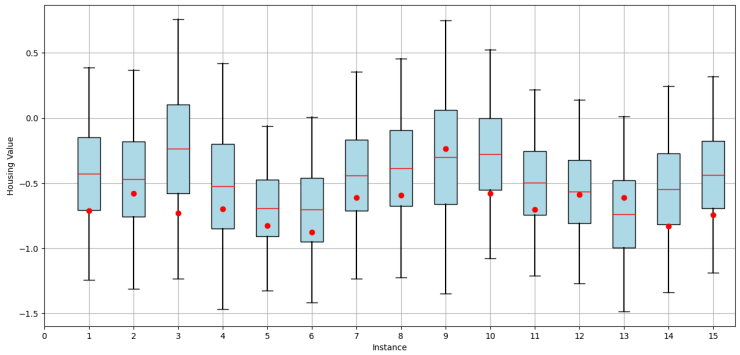

Figure 6: The red dots show the true median housing values (standardized and normalized), while the boxplots visualise the predictions of the proposed model for the 25% and 75% quantiles of the predicted distribution.

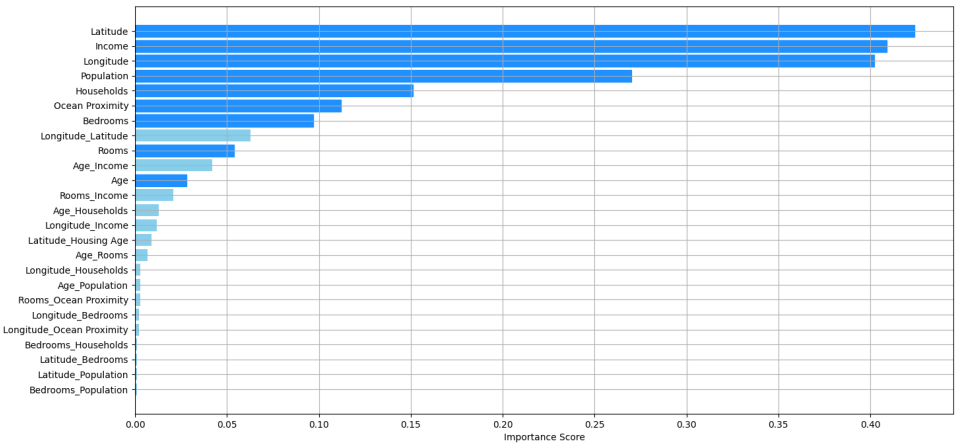

Figure 7: Individual and two way feature importances for the California housing dataset using the proposed model showing that the latitude, income and longitude are the most important features for predicting the house value.

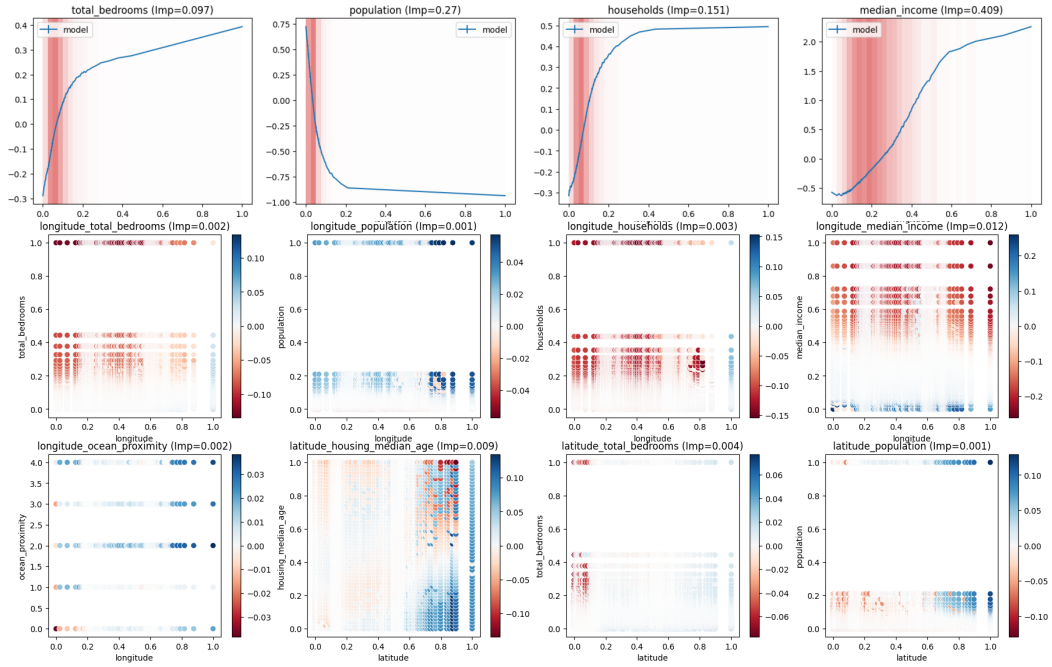

Figure 8: Shows the mean effects predicted by the model for the covariates and two way interaction effects. The model correctly captures the increase in housing value with an increase in the number of bedrooms, income and households and the negative effect with population along with the effect on two way interactions.

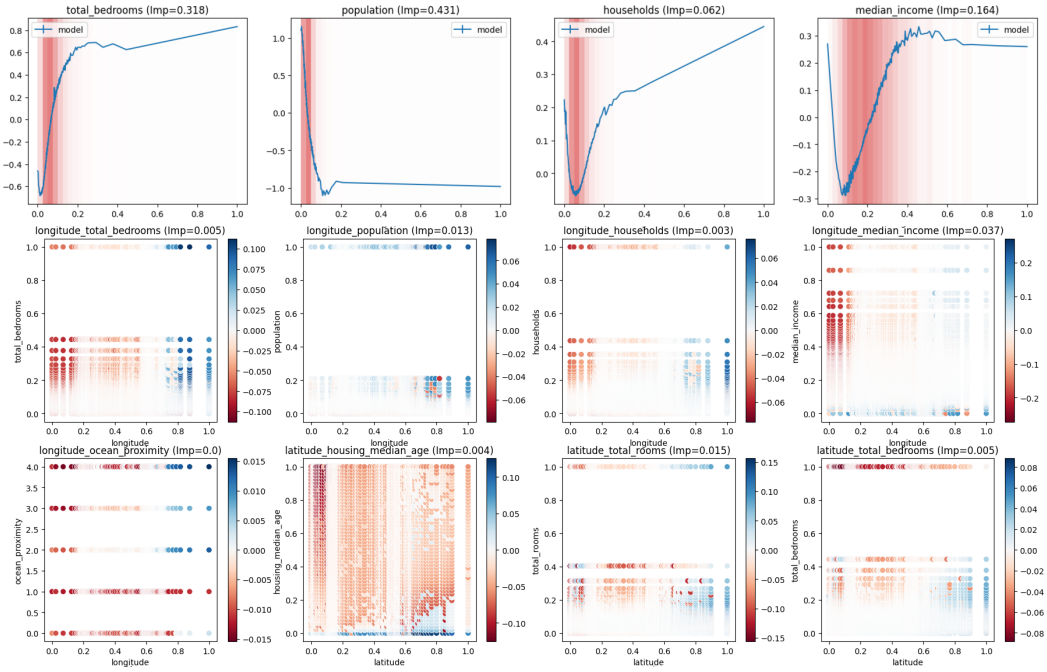

Figure 9: Shows the raw variance effects predicted by the model for the covariates and two way interaction effects. The model correctly captures a sharp decrease in variance with an increase in income before it increases again.