Generalizing Neural Additive Models via Statistical Multimodal Analysis

Anonymous authors
Paper under double-blind review

Abstract

Generalized Additive Models (GAM) Hastie (2017) and Neural Additive Models (NAM) Agarwal et al. (2021) have gained a lot of attention for addressing trade-offs between accuracy and interpretability of machine learning models. Yet, these models underperform when the data has multiple subpopulations with distinctive relationships between features and outputs. The main reason behind this limitation is that these models collapse multiple relationships by being forced to fit the data in a unimodal fashion. Here we propose a Mixture of Neural Additive Models (MNAM) to overcome these limitations. The proposed MNAM learns relationships between features and outputs in a multimodal fashion and assigns a probability to each mode. Based on a subpopulation, MNAM will activate one or more matching modes by increasing their probability. Thus, the objective of MNAM is to learn multiple relationships and activate the right relationships by automatically identifying subpopulations of interest. Similar to how GAM and NAM have fixed relationships between features and outputs, MNAM will maintain interpretability by having multiple fixed relationships. We demonstrate how the proposed MNAM balances between rich representations and interpretability with numerous empirical observations and pedagogical studies. The code is available at (to be completed upon acceptance).

1 Introduction

Deep neural networks (DNN) achieve extraordinary results across several important applications such as object detection Redmon et al. (2016); Girshick et al. (2014); Ren et al. (2015), object classification He et al. (2016); Krizhevsky et al. (2017); Dosovitskiy et al. (2020), and natural language processing Mikolov et al. (2013); Devlin et al. (2018); Brown et al. (2020). Yet DNN's popularity is still low in critical applications where miss-classification has high consequences or transparency is required for decision-making, e.g., to prevent unfairness toward certain groups; examples are medical-related risk estimation and machine learning (ML) based public policies. According to experts in these domains, one of the main factors limiting the adoption of DNN-based approaches is the lack of interpretability and trustworthiness associated with these algorithms Shorten et al. (2021); Amarasinghe et al. (2020); Li et al. (2022). Even though several techniques have been proposed to increase the understanding of DNN Agarwal et al. (2021); Ribeiro et al. (2016); Pedapati et al. (2020), medical professionals or policymakers still prefer simple models for which they can understand directly the factors that lead to a particular prediction. On the opposite end of DNN are algorithms such as linear regression and its multiple variants Montgomery et al. (2021), which are simple and interpretable but lack the flexibility and high performance that DNN has. Notably, linear models can't capture nonlinear relationships and can't exploit numerous novel tools that efficiently optimize modern DNN approaches. A recent approach proposed by Agarwal et al., named Neural Additive Models (NAM), provides an interesting balance between interpretability and learning power. Individual features undergo nonlinear transformations independently, and these transformed features are merged in a regression-like paradigm, allowing the user to understand the weight of each factor leading to a prediction. This enables the algorithm to learn non-trivial relationships between the features and the target outcomes while leveraging powerful state-of-the-art optimization tools developed for deep learning.

One of the main limitations of NAM is its lack of power to capture multimodal relationships between the input and the target variables. For example, imaging in the context of a medical application where we are predicting the glucose level y using electronic health records (EHR) as input variables x_1, ..., x_n; let us assume there are two subpopulations identified by the variable $d \in \{0, 1\}$, which can be observed or latent features. For both cases, NAM would fail to capture a relationship in which y is positively correlated with $(x_1|d = 0)$ but is uncorrelated with $(x_1|d = 1)$. This is due to NAM only learning one deterministic relationship between input and output. When d is a latent variable, NAM will fail to differentiate them and collapse two relationships into one by averaging them to learn one deterministic relationship. Even if d is an observed feature, NAM will fail to differentiate them as a DNN assigned for X_1 doesn't take d as an input to have information on two subpopulations.

To address this while preserving the virtues of NAM, we propose a probabilistic Mixture of Neural Additive Models (MNAM). The main idea is to model the relationship between input and outcome in a multimodal relationship and associate a probability to each mode. The probability of each mode enables the model to be flexible in representing multiple subpopulations as MNAM is able to activate accurate relationships for certain subpopulations by increasing their probability.

Figure 1 illustrates the power and flexibility of MNAM. These strengths are also illustrated in Section 3 through applying MNAM on real datasets. Such flexibility will be especially crucial in decision-making with high consequences. For example, for analyzing the side effects of medicine, 99% of participants might have steady glucose levels but 1% might have high and dangerous glucose levels after taking a medicine. NAM will collapse both levels into one indicating no side effects on average, but MNAM will accurately show, with probability, two glucose levels of different subpopulations.

It is important to highlight the interpretability of the model. Similar to NAM having a one fixed relationship between input features and output variables, MNAM will have fixed multiple relationships, which makes the model interpretable. Only the probability of each mode will change from the change in other features, which indicates changes in a subpopulation. Finally, just as for NAM, all powerful state-of-the-art tools developed for deep learning are applicable to MNAM.

Our main contributions are: (i) we propose a model called MNAM that could learn multiple relationships among subpopulations; (ii) we propose a method to train MNAM, with objectives to learn multiple relationships and activate one or more matching relationships for a given subpopulation; and (iii) we demonstrate MNAM is more expressive in accuracy and flexible in interpretability compared to NAM. We describe the proposed method in Section 2. Section 3 presents empirical evidence and pedagogical studies, showing strengths of MNAM. We discuss related work in Section 4 and limitations in Section 5. Finally, we provide a conclusion in Section 6.

2 Method

2.1 Architecture

In order to represent the multimodal relationship between inputs and outcomes, MNAM has an outcome of a mixture of k Gaussian distributions, which are described as $(\mathcal{N}_1(\mu_1,\, \sigma_1^2), ..., \mathcal{N}_k(\mu_k,\, \sigma_k^2), \pi_1, ..., \pi_k)$. $\mathcal{N}_i(\mu_i,\, \sigma_i^2)$ denotes the standard Gaussian distribution with μ_i as mean and σ_i as standard deviation, while π_i represents the probability associated with it. Since the Gaussian mixture model is a universal approximator for any density distributions, MNAM will be able to approximate any multimodal relationships given large enough k. We formalize this notion in Section 3.2. One or more Gaussian distributions will be assigned to one of the input-output relationships for the representation and MNAM will activate certain relationships for given subpopulations of the input by increasing the probability of the appropriate Gaussian distributions. This is an important property as it indicates that we can successfully capture and represent modes for relationships on various subpopulations in the dataset, without knowing the number of modes in advance. Such property will be shown in Section 3.2 through a pedagogical example.

Similar to NAM, MNAM predictions are built from a linear combination of embeddings Z_i of each input feature X_i mapped through a neural network. In contrast with NAM, MNAM embedding consists of pa-

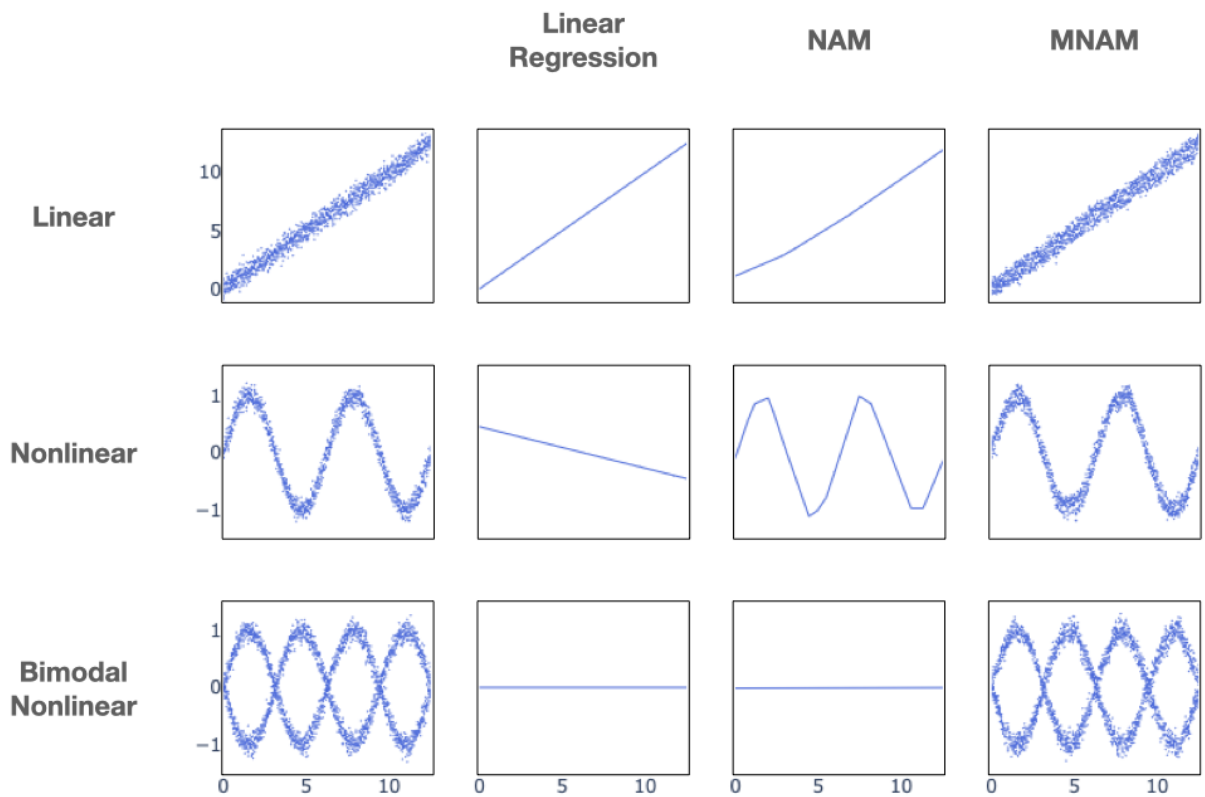

Figure 1: Linear regression, NAM, and MNAM on linear, nonlinear, and bimodal data. The left column illustrates the input for three datasets. The columns illustrate the representations learned by linear regression, NAM, and MNAM, respectively. As expected, linear regression fails to learn datasets with nonlinear relationships. NAM fails to learn datasets with relationships that have more than one modality, and only MNAM is able to learn nonlinear and multimodal relationships.

rameters for k Gaussian distributions and a latent variable for predicting the probability of the mixture of k Gaussian distribution models $(\mathcal{N}_{1,j}(\mu_{1,j},\, \sigma^2_{1,j}), ..., \mathcal{N}_{k,j}(\mu_{k,j},\, \sigma^2_{k,j}), Z^{\pi}_j)$. The left index of the Gaussian distributions is a reference to the number of components for the mixture and the right index j is a reference to one of the input features. As shown in Equation 1, we compute the mean and variance of the Gaussian distributions for the MNAM outcome by linearly combining the mean and variance of matching components for Gaussian distributions of features' embedding.

$$\begin{aligned} &\mathcal{N}_{i,1}(\mu_{i,1},\, \sigma^2_{i,1}) + ... + \mathcal{N}_{i,m}(\mu_{i,m},\, \sigma^2_{i,m}) \\ &= \mathcal{N}(\sum_{j=1}^{m} \mu_{i,j}, \sum_{j=1}^{m} \sigma^2_{i,j}) = \mathcal{N}_i(\mu_i,\, \sigma^2_i) \end{aligned} \tag{1}$$

The advantage of this linear property of summation for the Gaussian distributions is that MNAM is able to linearly represent how much the overall mean and uncertainty of prediction changes due to changes in a feature.

Latent variables for predicting the probability of the mixture of k Gaussian distributions for all features' embeddings will be the input for a separate neural network that predicts the probability of the output.

This neural network will learn to identify which subpopulation is being represented based on input from all features, and activate the correct relationships by assigning a high probability to the matching Gaussian distributions. The description and comparison of the architecture for NAM and MNAM is illustrated in Figure 2.

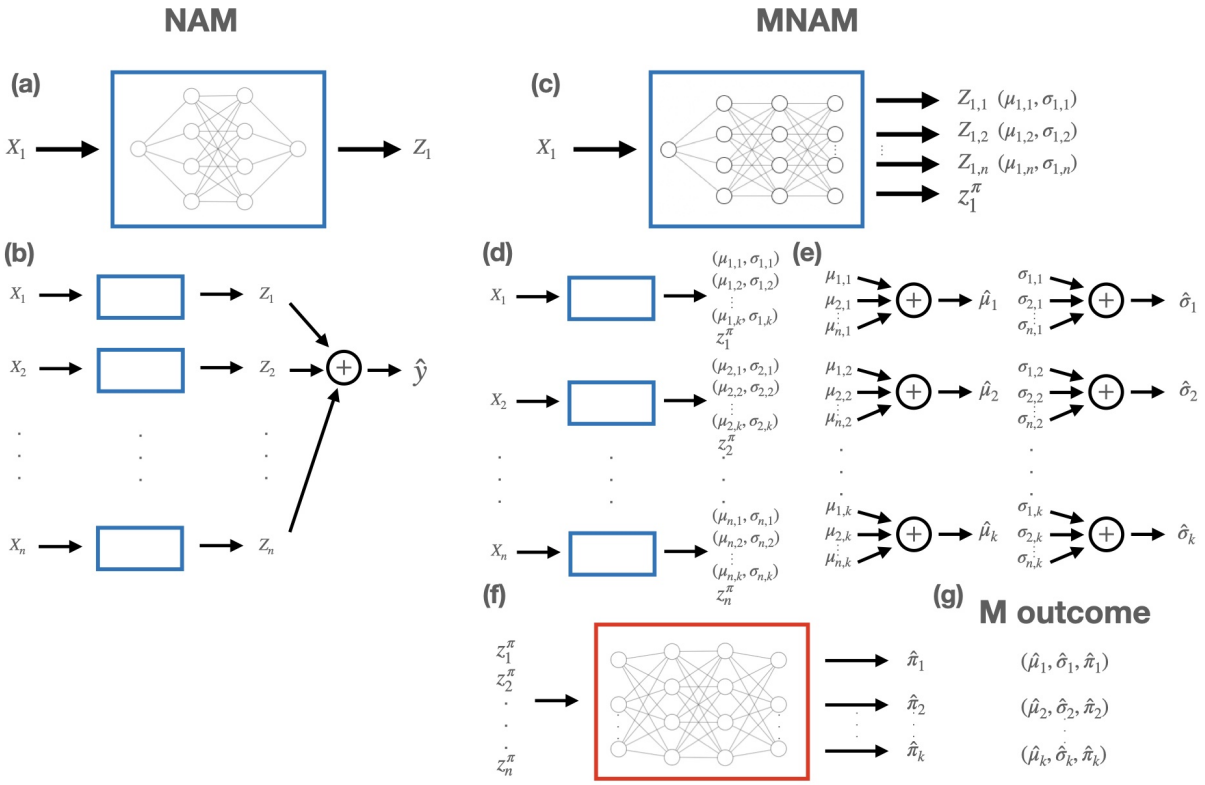

Figure 2: Illustrative schemes of NAM and MNAM network architectures. As shown in (a) and (b), NAM independently maps features into embedding through neural networks and then linearly combines embeddings for a prediction. Similar to NAM, MNAM independently maps features into embeddings through neural networks. The difference is that embedding consists of k Gaussian distributions and a latent variable for predicting probabilities for a mixture of the k Gaussian distributions, which is illustrated in (c) and (d). (e) illustrates linear combinations of each component of the Gaussian distributions for all features' embeddings. (f) depicts the mapping of latent variables for a mixture of k Gaussian distributions $(Z_1^\pi, Z_2^\pi, ..., Z_n^\pi)$ into probabilities for the mixture of k Gaussian distributions through a neural network. (g) is an example of the outcome for MNAM, which is the mixture of k Gaussian distributions.

2.2 Training and Optimization

As mentioned in Section 1, state-of-the-art optimization tools for deep learning are applicable for training MNAM. For this work, we used Adam Kingma & Ba (2014) with a learning rate decreasing by 0.5% for each epoch. The objective of the training and optimization of MNAM is to assign one or more Gaussian distributions to each relationship in the dataset. Another objective is to learn to identify subpopulations from the given features to activate the correct relationship associated with the given sample. We devise a hard-thresholding (HT) algorithm for the given objectives, these are described next. We also devise a soft-thresholding (ST) Algorithm. The description of the ST algorithm and comparison to the HT algorithm can be found in Section A in the appendix.

2.2.1 Hard-Thresholding (HT) Algorithm

Given the output of a mixture of k Gaussian distributions for MNAM, the Gaussian negative log-likelihood (GNLL) loss is computed for each Gaussian distribution against a label. Among k losses, only the minimum factor will be used to compute the total loss, which means only weights used to compute minimum loss are updated from a backpropagation. This enables the model to assign one Gaussian distribution to learn each relationship. Cross-entropy loss between the probabilities of a mixture of Gaussian distributions for a prediction and the index number of the Gaussian distribution with the minimum loss is computed to measure how well MNAM activates the corresponding Gaussian distribution for the input. This loss enables the model to learn to identify subpopulations for a given input to increase the probability of the correct Gaussian distribution for representation. Algorithm 1 summarizes the proposed training algorithm. It is important to highlight that the proposed learning method is unsupervised, in the sense that the data subgroups do not need to be known or defined in advance.

Algorithm 1 Hard-Thresholding (HT) Algorithm

Input: Data (X, Y), MNAM f, GNLL loss g, Cross-entropy loss function h, Rate for cross-entropy loss λ
$\mathcal{N}_1(\mu_1, \sigma_1), ..., \mathcal{N}_k(\mu_k, \sigma_k), \pi_1, ..., \pi_k = f(X)$
$min_loss = 0$
for $i = 1$ **to** k **do**
  $gau_loss = g(\mathcal{N}_i(\mu_i, \sigma_i), Y)$
  if $min_loss > gau_loss$ **then**
    $min_loss = gau_loss$
    $min_index = i$
  end if
end for
$prob_loss = h((\pi_1, ..., \pi_k), min_index)$
$total_loss = Min_loss + \lambda \cdot prob_loss$

2.3 Regularization

Similar to NAM, all regularization methods for deep learning can be applied to MNAM, including weight decay, dropout, and output penalty. For this study, we utilized weight decay and output penalty.

3 Result

3.1 Empirical Observations

3.1.1 Datasets

We evaluate six datasets: the California Housing (CA Housing) Pace & Barry (1997), the Fair Isaac Corporation (FICO) FICO (2018), the New York Citi Bike (BIKE) Vanschoren et al. (2013), the Medical Information Mart for Intensive Care (MIMIC-III) Johnson et al. (2016), the US Census data on Income (ACS Income) for California Ding et al. (2021), and the US Census data on Travel time (ACS Travel) for California Ding et al. (2021).

CA Housing: The CA Housing dataset has the task of predicting housing prices and it consists of eight features.

FICO: The FICO dataset has the task of predicting credit scores and it consists of 24 features.

BIKE: The BIKE dataset has the task of predicting the duration of trips and it consists of four features. Due to limited computation resources, we dropped data points that had more than 4000 seconds of duration for a bike trip and sampled 25% of the remaining dataset for analysis.

MIMIC-III: MIMIC-III dataset has the task of predicting the length of hospitalization for patients and it consists of various static and dynamic features. For comparing NAM and MNAM, we have used only static features, which consist of seven features.

ACS Income: The ACS Income dataset has the task of predicting income and it consists of ten features.

ACS Travel: The ACS Travel dataset has the task of predicting travel time to work and it consists of 16 features.

3.1.2 Training and Evaluation

Similar to how the original paper trained NAM, we used Bayesian optimization Močkus (1975) to finetune variables to train NAM and MNAM. Learning rate, weight decay, and output penalty are finetuned for NAM. Learning rate, weight decay, output penalty, number of Gaussian distributions, and lambda for cross-entropy loss are finetuned for MNAM. Optimized parameters from Bayesian optimization can be found in the table from Section B in the appendix. We used a 5-fold cross-validation for CA Housing, FICO, and MIMIC-III datasets, and a 3-fold cross-validation for BIKE, ACS Income, and ACS Travel datasets. For evaluation, we trained 20 different models by randomly splitting the train set into train and validation sets for each fold. We ensembled 20 models to evaluate on the test set.

For comparison, we consider mean absolute error (MAE) and earth mover's distance (EMD) as evaluation metrics. We used MAE to assess the overall accuracy in the mapping of features to output for MNAM, and used EMD to evaluate how well the model's output represents the distribution of labels for the test dataset.

3.1.3 Results

Table 1 shows the MAE and EMD scores of NAM and MNAM on datasets described above. MNAM had similar or better scores in MAE compared to NAM in all six datasets. For the EMD score, MNAM had a significantly better performance compared to NAM in all six datasets. This shows that in addition to learning suitable mappings, MNAM is significantly better at learning the distribution of the output compared to NAM.

Differences in performance between MNAM and NAM differ greatly by datasets. For example, the difference in the EMD score between NAM and MNAM for the BIKE dataset is much more significant than for the FICO dataset. There could be multiple explanations for such phenomena. It could be due to the FICO dataset having less uncertainty or variances for labels. Another explanation could be due to the FICO dataset having less number of modes. An example of this is shown and illustrated in Figure 1. In that case, MNAM would improve the EMD score by the number of modes and complexity of a dataset

	NAM		MNAM	
Dataset	MAE	EMD	MAE	EMD
CA Housing	$0.48\pm 9e^{-05}$	$0.24 \pm 6e^{-05}$	$0.46 \pm 4e^{-05}$	0.077 ± 0.0001
FICO	2.7 ± 0.002	0.73 ± 0.01	2.7 ± 0.002	0.60 ± 0.005
MIMIC	1.5 ± 0.0002	1.43 ± 0.0004	1.5 ± 0.0003	$0.24 \pm 2e^{-05}$
BIKE	3.4 ± 0.0005	2.50 ± 0.0003	3.4 ± 0.0006	0.26 ± 0.0002
ACS Income	37.2 ± 0.003	21.3 ± 0.1	35.7 ± 0.02	7.4 ± 0.04
ACS Travel	15.6 ± 0.0004	12.8 ± 0.003	15.5 ± 0.002	3.1 ± 0.02

Table 1: MAE and EMD score for NAM and MNAM on CA Housing, FICO, MIMIC, BIKE, ACS Income, and ACS Travel dataset

3.1.4 Interpretability

In this section, we visualize the relationships between features and labels, and how different relationships are activated from changes in subpopulations; we illustrate this for the CA Housing dataset. This illustrates the strength of the interpretability of MNAM. Relationships plots for other datasets can be found in Section C

in the appendix. As illustrated in Figure 3, MNAM is able to learn and represent multiple relationships between features and labels, which NAM fails to do as it collapses those relationships into a mean. Therefore, MNAM is more flexible in explaining and representing multiple relationships between features and labels by activating one or multiple of them.

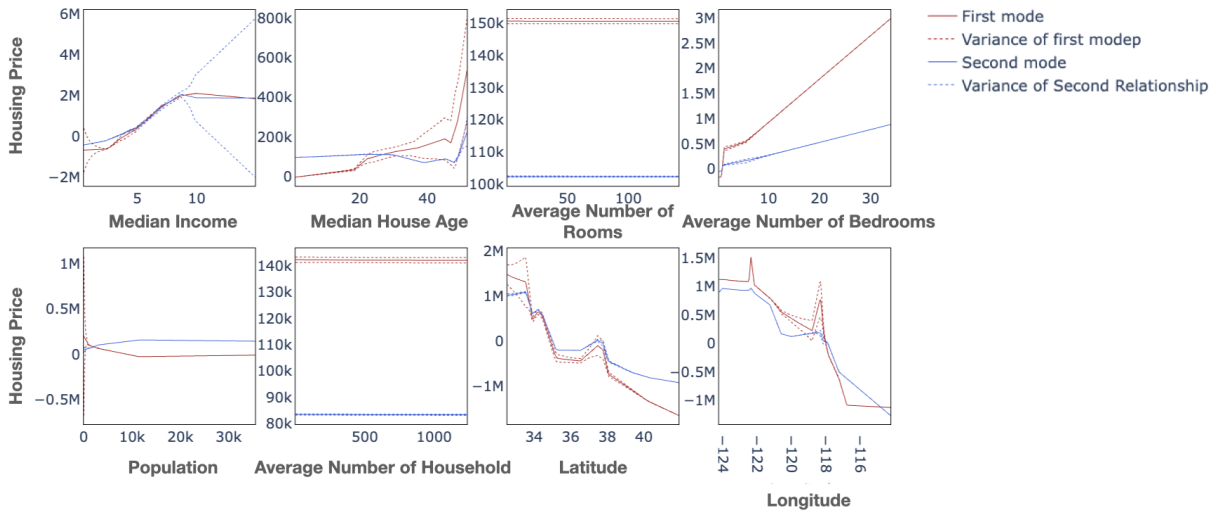

Figure 3: Two relationships between features and label learned by MNAM on CA Housing dataset. Solid lines represent the mean of relationships and dotted lines represent the uncertainties of relationships.

Allowing multimodal data representations sheds light on non-trivial data relationships that are otherwise hidden in average “one-fit-all" models. For example, as illustrated in Figure 4, we identified that the price of a house could increase or decrease as the number of people in the neighborhood increases (the first column of Figure 4, illustrates the two modes recognized by MNAM). If we group the algorithm's output by median income (the first row of the second column represents the bottom one percent, and the first row of the third column represents the top one percent), we can recognize that one of the modalities is associated with higher income households and the other with lower income households. For example, the first row of the second column shows that the top mode is activated more frequently on this subgroup (darker blue represents higher frequency), suggesting that the larger the number the people in the neighborhood, the higher the house prices. The opposite can be recognized for the higher-income subgroups (see the first row of the third column). In other words, the output of the model suggests that for wealthier neighborhoods, the more people, the less expensive houses are, while the opposite occurs in poor communities. A similar story is illustrated when we group the algorithm's output by proximity to the beach (the second row of the second column represents inland, and the second row of the third column represents the area near the beach). The output of the model suggests that for areas near beaches, the more people the more expensive houses are, while the opposite occurs inland. Notice how these rich data interpretations would have been missed using NAM, where a “one-fit-all" model is optimized.

3.2 Pedagogical Example

For pedagogical value and to further illustrate the differences between the original NAM and the proposed MNAM, we created a synthetic dataset with different subpopulations, which are differentiated by either observed or latent variables. NAM has limitations in accurately representing such dataset as it collapses four relationships between X_1 and Y into one deterministic relationship by averaging them. When X_2 is an observed variable, NAM is not able to differentiate relationships, since a neural network assigned to X_1 does not take X_2 as input. The neural network for X_1 simply uses the average relationship for representation, which is shown when $X_2 = 0$ and $X_2 = 1$. The representation is worsened, when $X_2 = 2$

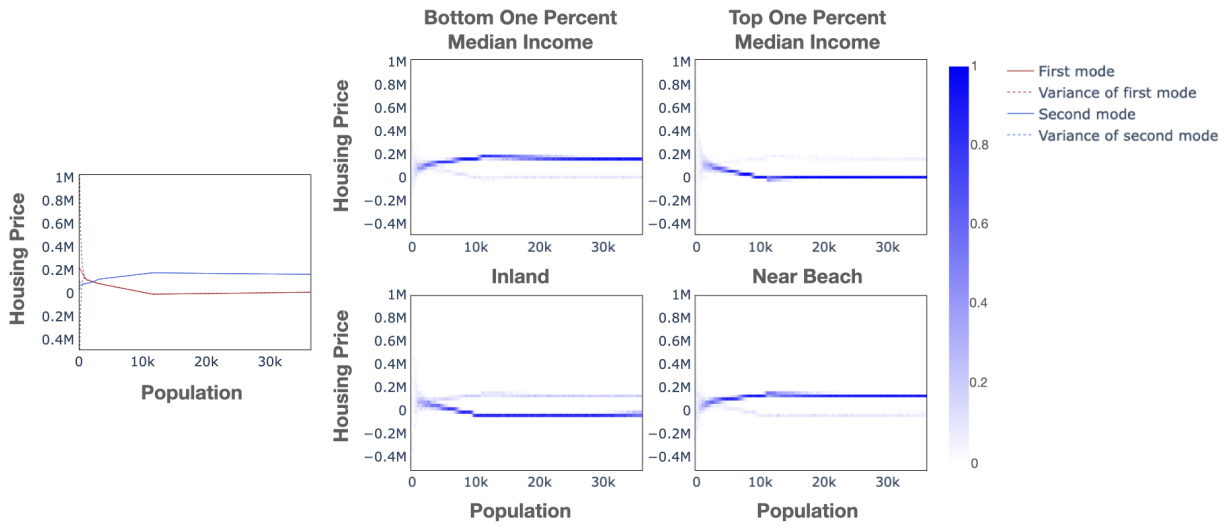

Figure 4: Line graph and heatmaps on the relationship between housing price and population for CA Housing. The first column, a line graph, represents two modes recognized by MNAM between housing prices and populations. Second and third columns, heatmaps, represents changes in the activation of two modes from changes in other remaining features. The first row represents changes in median income and the second row represents changes in proximity to the beach. The magnitude of the mode's activation is illustrated through the intensity of color in heatmaps. Darker blue represents higher activation of a mode. The blue color bar represents the magnitude of a mode's activation.

and $X_2 = 3$, as it tries to represent multiple relationships with one relationship, as shown in the second column of Figure 5. MNAM overcomes such limitations as it is able to learn four relationships and activate the right relationships for each subpopulation. Another strength of MNAM is that as long as k is larger than the number of relationships in a dataset, MNAM will be able to represent the relationships accurately. In other words, tuning k is not critical, as long as its value is higher than the expected number of modes. Furthermore, MNAM is able to learn the uncertainty of each relationship, which NAM is unable to do. Described limitations of NAM and strengths of MNAM are illustrated in Figure 5.

3.3 Trade-offs between Accuracy and Interpretability

In this section, we compared different models to explore trade-offs between accuracy and interpretability. We evaluated Linear Regression (LR); NAM; the here proposed MNAM; Explainable Boosting Machine (EBM) Nori et al. (2019), which is a form of Generalized Additive Models (GAM) with pairwise interaction terms; and Gradient Boosting Trees (GBT) Friedman (2001); Pedregosa et al. (2011). We used grid search for LR, EBM, and GBT to finetune hyperparameters for training. Table 2 shows the MAE and EMD scores for these five models. The order of the columns, left to right, represents an increase in complexity and a decrease in interpretability (here considered as a clear relationship between input and output). The table is split into two, which are models with direct relationships and complex relationships (left and right respectively). LR, NAM, and MNAM are models with direct relationships because their feature and output relationships are fixed even from changes in other features. Meanwhile, EBM and GBT are considered as models with complex relationships as their feature and output relationships changes from a change in other features due to their interaction terms. With this complexity, it becomes difficult to interpret those models.

Even though the MAE score improves from an increase in the complexity of models for most datasets (as expected), differences in performances among models fluctuate greatly by datasets. This could be a result of datasets having different complexity. For example, models have similar performances on the MIMIC datasets. This could be due to the datasets being too simple to not even require nonlinearity or interaction

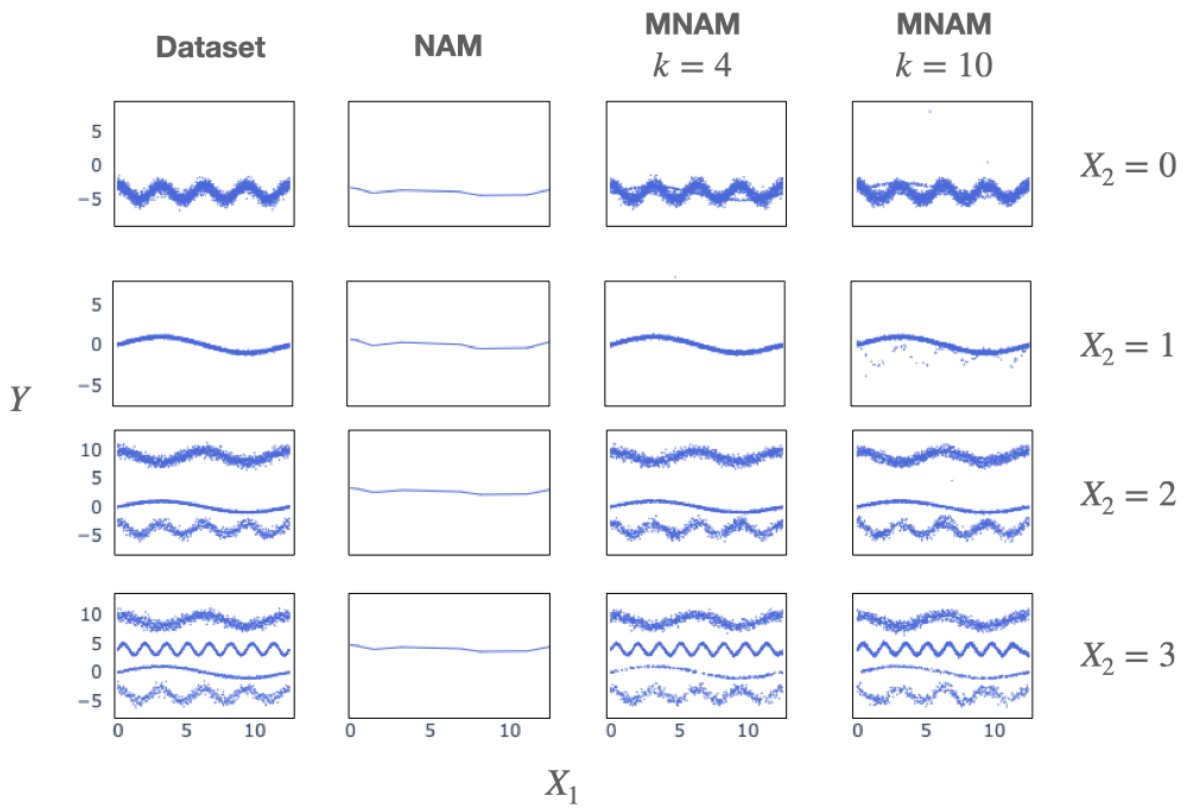

Figure 5: NAM versus MNAM on a dataset that has variables that identify subpopulations as observed and latent variables. The left column is a scatter plot for a dataset with different values of X_2. The remaining columns represent predictions from training on the dataset for NAM, MNAM with $k = 4$, and MNAM with $k = 10$. NAM clearly fails to represent the dataset as it collapses multiple relationships into one relationship. On contrary, MNAM with $k = 4$ and $k = 10$ accurately represents the dataset as it learns four relationships and activates the right ones for different values of X_2.

terms of models for representations. In contrast, for the ACS Income dataset, the performance increases with an increase in complexity. This could be due to the dataset being more complex and requiring nonlinearity and more interaction terms with higher degrees for models to represent the dataset well.

Similar to MAE, the EMD score improves through an increase in the complexity of models except for the proposed MNAM. MNAM outperforms all models in most of the datasets. As mentioned in Section 3.1.3, the possible explanation is that MNAM is the only model that learns the uncertainty and multimodality of datasets compared to other models.

4 Related Works

For interpretable models, GAM Hastie (2017) has been widely used. GAM transforms each feature by a function and linearly combines the transformed features, which enables features to have a fixed relationship with the output. NAM Agarwal et al. (2021) uses neural networks while GAM uses boosted decision trees Lou et al. (2012); Guisan et al. (2002) to transform the features. Compared to those models, MNAM has multiple outputs with probability, instead of one single estimate. These multiple outputs enable the model to represent multiple subpopulations in the dataset. Furthermore, it is more flexible for interpretation as it is

	Direct Input and Output Relationships						Complex Input and Output Relationships			
	LR		NAM		MNAM		EBM		GBT	
Dataset	MAE	EMD	MAE	EMD	MAE	EMD	MAE	EMD	MAE	EMD
CA Housing	0.54	0.29	0.48	0.24	0.46	0.077	0.34	0.11	0.31	0.09
FICO	3.38	1.16	2.7	0.73	2.7	0.60	2.5	0.51	2.4	0.51
MIMIC	1.5	1.36	1.5	1.43	1.5	0.24	1.5	1.32	1.5	1.25
BIKE	3.65	3.06	3.4	2.50	3.4	0.26	3.4	2.45	3.4	2.43
ACS Income	40.0	27.1	37.2	21.3	35.7	7.4	33.3	14.5	31.8	12.9
ACS Travel	16.8	14.2	15.6	12.8	15.5	3.1	14.2	8.9	13.8	8.7

Complexity →

Interpretability ←

Table 2: MAE and EMD score for LR, NAM, MNAM, EBM, and GBT on CA Housing, FICO, MIMIC, BIKE, ACS Income, and ACS Travel datasets. The complexity of models increases from left to right and the interpretability of models increases from right to left.

able to show multiple relationships between features and labels, and how different relationships are activated by changes in a subpopulation.

To address the limitation of GAM in representing multiple subpopulations in a dataset, Generalized Additive Model with Pairwise Interactions (GA2M) Karatekin et al. (2019) or EBM has been proposed, which adds interaction terms into GAM. Yet, the limitation of GA2M is that relationships between features and labels are not fixed due to its interaction terms, making the model less interpretable. When more interaction terms are added, the model becomes less interpretable. Compared to GA2M, MNAM has k fixed relationships between features and labels, which makes it interpretable, and the only changes are in the activation of relationships from the changes in features.

Mixture Density Network (MDN) Bishop (1994) is the first model to use a mixture of k Gaussian distributions as an outcome for a neural network. Its purpose was to solve inverse and robotics problems. MDN is not a form of Generalized Additive Model but more of a neural network model with a mixture of k Gaussian distributions as an outcome. Thus, similar to a neural network, it does not have a fixed relationship between features and labels, and therefore it is not easily interpretable.

5 Limitations

MNAM's current formulation is only applicable to regression problems. Unlike continuous variables, binary variables are meaningless to cluster as the only possible values are zero and one. For our future work, we will utilize different algorithms such as local interpretable model-agnostic explanations (Lime) Ribeiro et al. (2016) to overcome such a limitation. For example, we could utilize MNAM to approximate predictions of a neural network that has been trained for the classification, as a prediction for the classification will be continuous. Using MNAM to approximate the prediction of the classification model, we will able to show multiple relationships between features and outputs and how those relationships are activated from changes in subpopulations or features.

MNAM trade-offs between the accuracy and interpretability of a model. Increasing the number of k Gaussian distributions for MNAM will increase accuracy. Yet, if the number of k Gaussian distributions is large, then it will be hard to interpret as there are too many possible relationships between features and outputs. The larger the number of k Gaussian distributions in MNAM, the more the model will become similar to neural networks as it covers all separate relationships for all possible combinations of features. For our future works, we would explore different penalties for the number of k Gaussian distributions in training to find an optimal balance between accuracy and interpretability.

6 Conclusion

In this work, we introduced Mixture Neural Additive Model (MNAM), an interpretable model with more flexibility compared to GAM and NAM. While GAM and NAM have only one estimate for an output and one relationship between features and outputs, MNAM has k multiple estimates for an output, with probability, and k relationships between features and outputs, to represent different relationships for each potential subpopulation separately. With such advantages in flexibility, we have shown that MNAM outperforms NAM in various datasets. Furthermore, we have shown how MNAM improves interpretation by illustrating how different relationships are activated by changes in subpopulations.

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.

Kasun Amarasinghe, Kit Rodolfa, Hemank Lamba, and Rayid Ghani. Explainable machine learning for public policy: Use cases, gaps, and research directions. *arXiv preprint arXiv:2010.14374*, 2020.

Christopher M Bishop. Mixture density networks. *Aston University*, 1994.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34:6478–6490, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

FICO. Fico explainable machine learning challenge, 2018. `https://community.fico.com/s/explainable-machine-learning-challenge`.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pp. 1189–1232, 2001.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

Antoine Guisan, Thomas C Edwards Jr, and Trevor Hastie. Generalized linear and generalized additive models in studies of species distributions: setting the scene. *Ecological modelling*, 157(2-3):89–100, 2002.

Trevor J Hastie. Generalized additive models. In *Statistical Models in S*, pp. 249–307. Routledge, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

Tamer Karatekin, Selim Sancak, Gokhan Celik, Sevilay Topcuoglu, Guner Karatekin, Pinar Kirci, and Ali Okatan. Interpretable machine learning in healthcare through generalized additive model with pairwise interactions (ga2m): predicting severe retinopathy of prematurity. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pp. 61–66. IEEE, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, pp. 1–38, 2022.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 150–158, 2012.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pp. 400–404. Springer, 1975.

Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

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

Tejaswini Pedapati, Avinash Balakrishnan, Karthikeyan Shanmugam, and Amit Dhurandhar. Learning global transparent models consistent with local contrastive explanations. *Advances in Neural Information Processing Systems*, 33:3592–3602, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Deep learning applications for covid-19. *Journal of Big Data*, 8(1):1–54, 2021.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL `http://doi.acm.org/10.1145/2641190.264119`.

A Other Training Algorithms

A.1 Soft-Thresholding Algorithm

Similar to the EM algorithm Dempster et al. (1977), the ST algorithm has expectation and maximization steps for training. In the expectation step, we compute the posterior probability of subpopulations $P(Z = k|X, Y)$. As shown in Equation 2, we compute the posterior probability by utilizing Bayesian Theorem,

$$\begin{aligned} P(Z = k|X, Y) &= \frac{P(X, Y|Z = k)P(Z = k)}{P(X, Y)} \\ &= \frac{P(X, Y|Z = k)P(Z = k)}{\sum_{i=1}^{k} P(X, Y|Z = k)P(Z = k)}, \end{aligned} \tag{2}$$

where $P(X, Y|Z = k)$ is the likelihood of kth Gaussian distribution for the giveninput, and $P(Z = k)$ is the prior probability of a subpopulation, which is predicted from MNAM. In the maximization step, we update the weights of MNAM to maximize the expectation or posterior probability of the subpopulations. First, we compute GNLL losses for all Gaussian distributions, and then GNLL losses for all the Gaussian distributions are linearly combined with weights matching posterior probabilities from the expectation step. This ensures weights used to compute Gaussian distribution with a higher likelihood are updated more. Cross-entropy loss between the prior probability predicted from MNAM and the posterior probability computed in the expectation step is computed with a similar purpose as in the HT algorithm. Algorithm 2 summarizes the proposed training algorithm.

Algorithm 2 Soft-Thresholding (ST) Algorithm

Input: Data (X, Y), MNAM f, GNLL loss g, Crossentropy loss function h, Rate for cross-entropy loss λ
$\mathcal{N}_1(\mu_1, \sigma_1), ..., \mathcal{N}_k(\mu_k, \sigma_k), \pi_1, ..., \pi_k = f(X)$
for $i = 1$ **to** k **do**
  $gau_loss_i = g(\mathcal{N}_i(\mu_i, \sigma_i), Y)$
  $gau_like_i = p(Y; \mu_i, \sigma_i)$
end for
$mar_prob = \sum_{j=1}^{k} gau_like_j \cdot \pi_j$
$\hat{\pi}_1, ..., \hat{\pi}_k = \frac{gau_like_1 \cdot \pi_1}{mar_prob}, ..., \frac{gau_like_k \cdot \pi_k}{mar_prob}$
$gau_loss = \sum_{i=1}^{k} gau_loss_i \cdot \hat{\pi}_i$
$prob_loss = h((\pi_1, ..., \pi_k), (\hat{\pi}_1, ..., \hat{\pi}_k))$
$total_loss = gau_loss + \lambda \cdot prob_loss$

A.2 Comparison of Training Algorithms

For comparing HT and ST algorithms, we evaluated numerical stability (NS), computation of time (CT), and accuracy. Using the dataset from the pedagogic study, we trained MNAM with different learning rates 20 times each to evaluate metrics. NS was assessed by computing the percentage of successful training without exploding gradient. CT was assessed by tracking average training time in seconds. Accuracy was assessed by computing MAE and EMD on the test set. Table 3 shows the evaluation of those metrics.

The HT algorithm had better performance in NS and CT. One of the explanations for better performance in NS is that the HT algorithm only passes minimum GNLL loss while the ST algorithm passes all GNLL losses with weights for an update. The ST algorithm passes more loss compared to the HT algorithm, which makes it numerically unstable during training. Furthermore, the ST algorithm has higher CT compared to the HT algorithm because it requires more computation to estimate the posterior probability, the HT algorithm only needs to find a minimum GNLL loss for training. For accuracy, the HT algorithm had a higher EMD score

and lower MAE score compared to the ST algorithm. Based on the priority of two metrics, one could choose one algorithm over the other. For this study, we used the HT algorithm due to its better performance in NS and CT.

	Hard-Thresholding Algorithm				Soft-Thresholding Algorithm			
LR	NS	CT	MAE	EMD	NS	CT	MAE	EMD
0.05	100%	217.61	43.89	145.38	0%	NA	NA	NA
0.01	100%	386.13	5.23	4.03	0%	NA	NA	NA
0.005	100%	470.94	3.12	0.25	0%	NA	NA	NA
0.001	100%	462.05	3.14	0.19	95%	488.77	2.82	0.40
0.0005	100%	929.62	3.09	0.19	100%	891.53	2.98	0.36
0.0001	100%	1029.1	3.12	0.29	100%	1036.5	3.06	0.30
$5e^{-05}$	100%	992.03	3.31	0.28	100%	1050.7	3.08	0.45

Table 3: Comparision of HT algorithm and ST algorithm on data from pedagogic study

B Table of optimized parameters for MNAM

Dataset	Learning rate	Weight decay	Output penalty	Number of Gaussian distribution	Cross-entropy loss
CA Housing	0.009896	3.8512e-05	0.03363	2	0.6118
FICO	0.02757	6.6649e-05	0.006145	6	0.4718
MIMIC	0.01805	7.0946e-05	0.01908	2	0.7214
BIKE	0.01172	9.1022e-05	0.09256	6	0.3537
ACS Income	0.02873	9.13e-05	0.00167	4	0.494
ACS Travel	0.01894	9.3377e-05	0.0028	4	0.3634

Table 4: Optimized parameters for MNAM on six datasets

C Relationships plots on other datasets

C.1 FICO

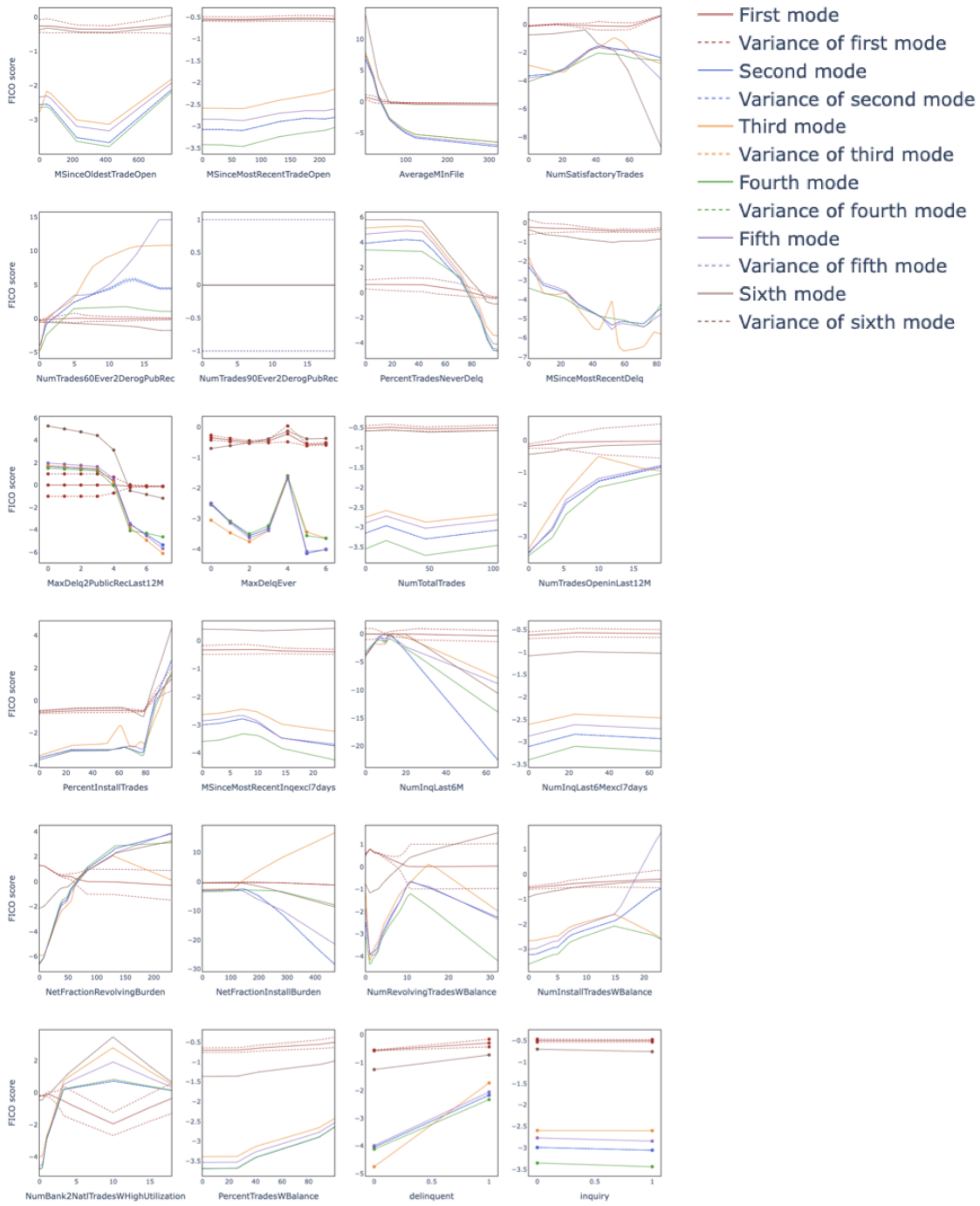

Figure 6: Learned relationships between features and labels for the MNAM on FICO datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

C.2 MIMIC

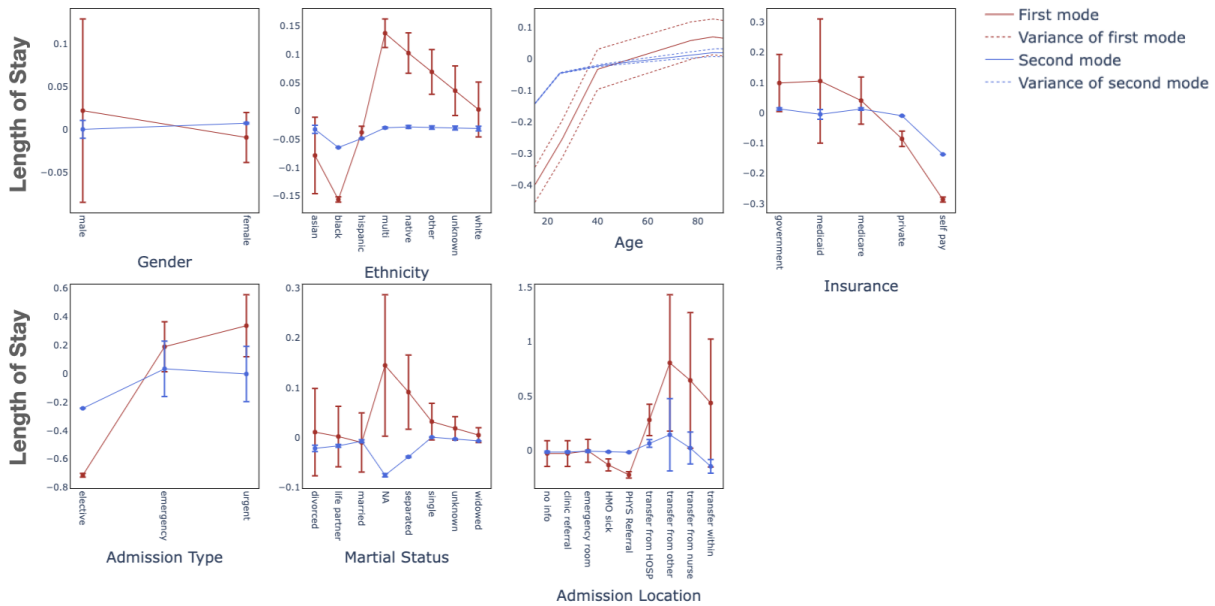

Figure 7: Learned relationships between features and labels for the MNAM on MIMIC datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

C.3 BIKE

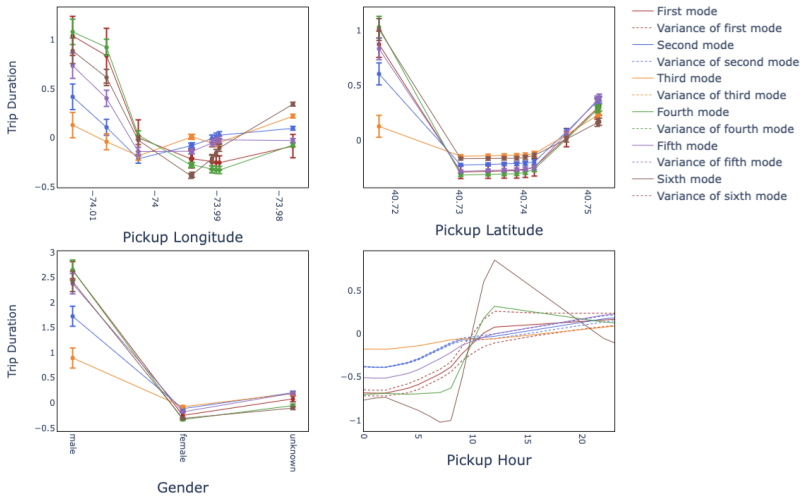

Figure 8: Learned relationships between features and labels for the MNAM on BIKE datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

C.4 ACS Income

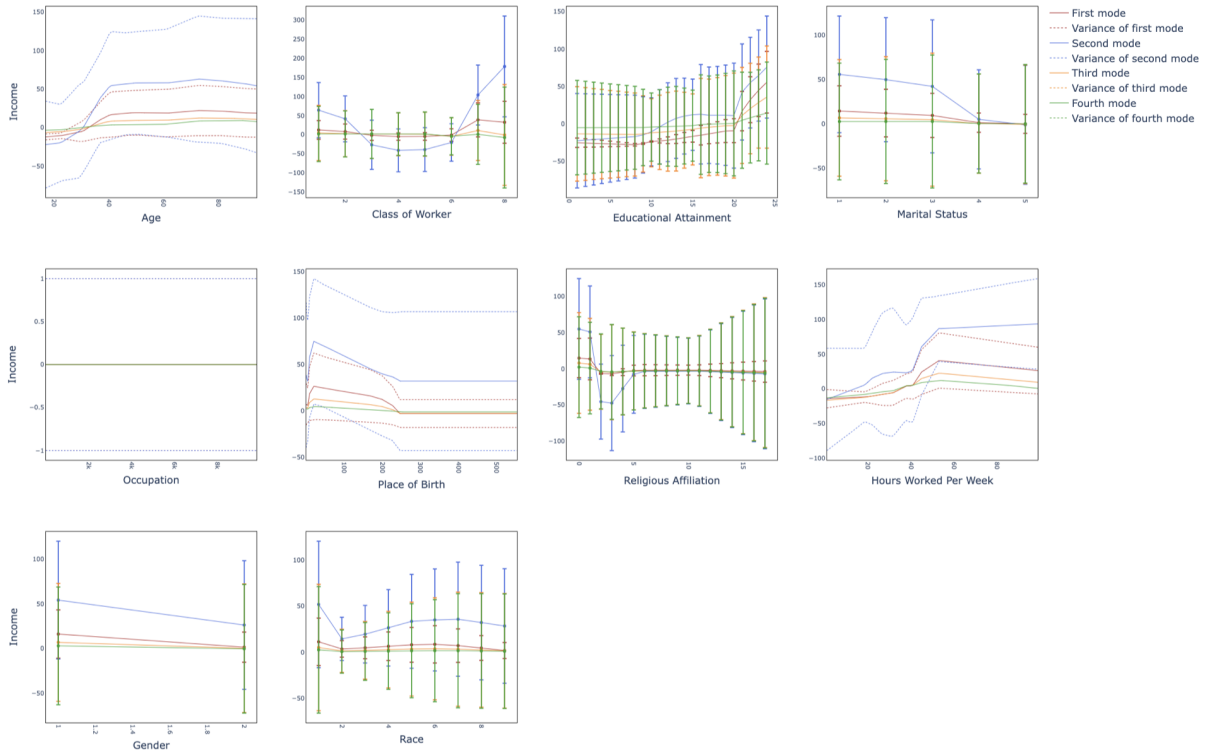

Figure 9: Learned relationships between features and labels for the MNAM on ACS Income datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

C.5 ACS Travel

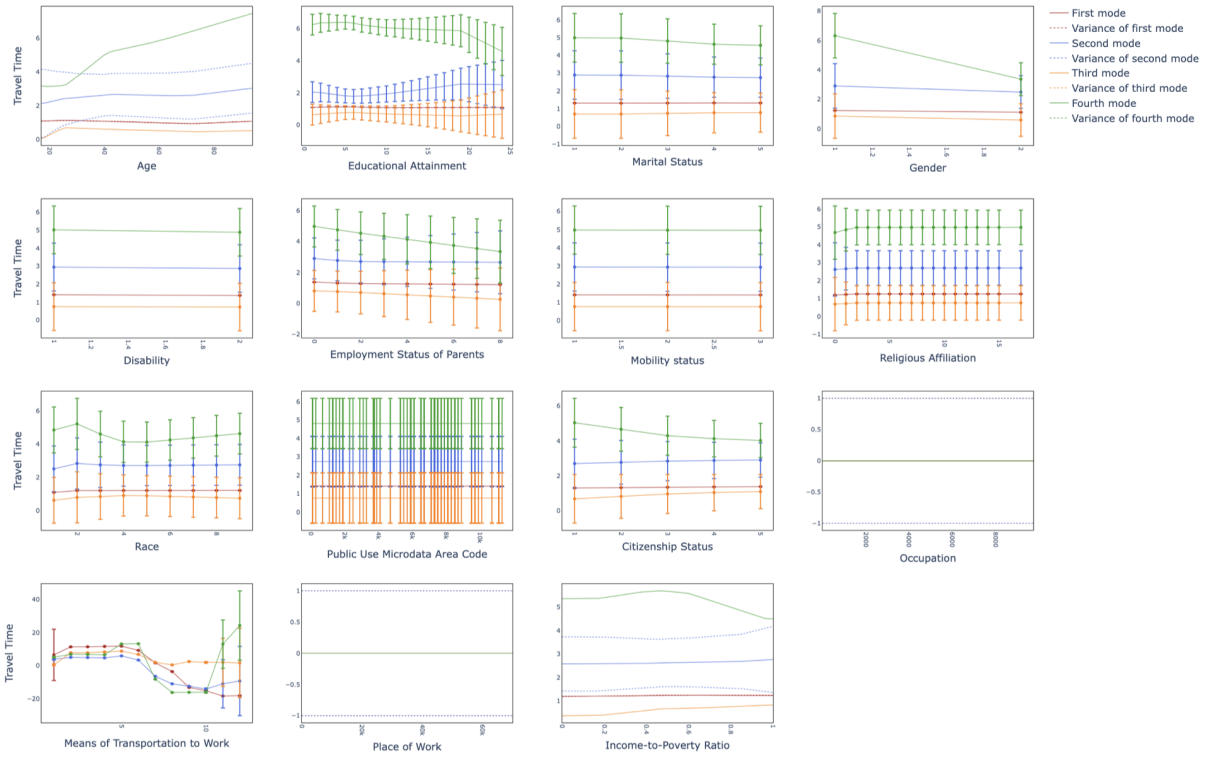

Figure 10: Learned relationships between features and labels for the MNAM on ACS Travel datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.