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. Although the field has focused on minimizing trade-offs between accuracy and interpretability, the limitation of GAM or NAM on data that has multiple subpopulations, differentiated by latent variables with distinctive relationships between features and outputs, has rarely been addressed. 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 address and describe the overlooked limitation of "one-fits-all" interpretable methods and propose a Mixture of Neural Additive Models (MNAM) to overcome it. 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), which is a form of Generalized Additive Models (GAM), 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.

Although most of the research on GAM has focused on minimizing the trade-offs between accuracy and interpretability Nori et al. (2019); Zuur (2012); Agarwal et al. (2021), addressing the lack of power for GAM and NAM in capturing multimodal relationships between input and target variables has been rare or nonexistent. This limitation is crucial especially when a dataset has multiple distinctive relationships between features and outputs. 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 apply mixture density networks (MDNs), a neural network with mixture of k Gaussian distributions as an outcome, as a linking function for GAM 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 identify the overlooked limitation that GAM and NAM have when they are trained with a dataset that has multiple subpopulations; (ii) we provide a practical alternative to solve the critical problem of "one-fits-all" standard in interpretable DL approaches; (iii) we propose a model called MNAM that could learn multiple relationships among subpopulations for the solution; (iv) 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 (v) 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

MNAM produces an outcome that is a mixture of k Gaussian distributions. These distributions can be represented as $(\mathcal{N}_1(\mu_1, , \sigma_1^2), ..., \mathcal{N}_k(\mu_k, , \sigma_k^2), \pi_1, ..., \pi_k)$. Here, $\mathcal{N}_i(\mu_i, , \sigma_i^2)$ represents a standard Gaussian distribution with mean μ_i and standard deviation σ_i, while π_i denotes the associated probability. The purpose of utilizing a mixture of k Gaussian distributions, a universal approximator for any density distribution,

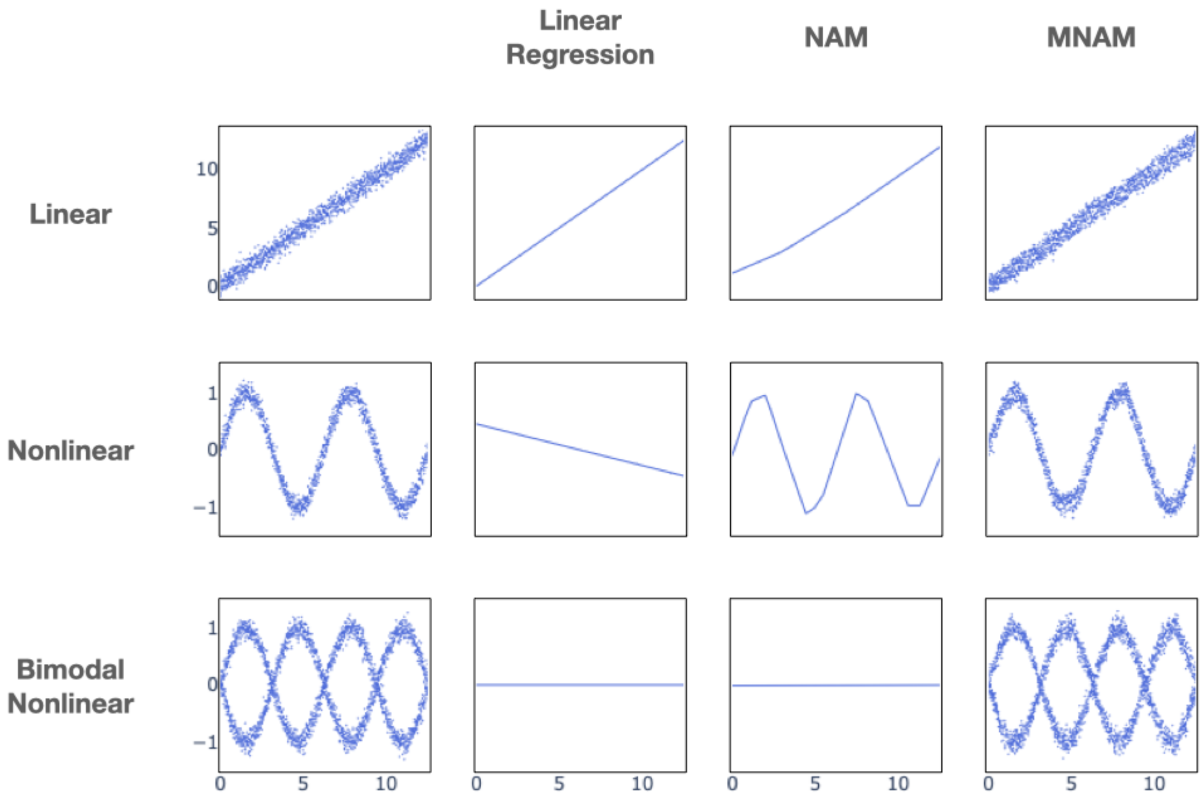

Figure 1: Linear regression, NAM, and MNAM on linear, nonlinear, and bimodal data. The left column illustrates the input for three datasets. The columns illustrate the representations learned by linear regression, NAM, and MNAM, respectively. As expected, linear regression fails to learn datasets with 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.

as the outcome for the model, is to represent multiple relationships between inputs and outcomes. For representation, one or more Gaussian distributions are assigned to each input-output relationship, and MNAM activates specific relationships for given subpopulations of the input by increasing the probability of the appropriate Gaussian distributions. This property is significant because it allows us to successfully capture and represent modes for relationships within various subpopulations in the dataset. Importantly, this approach does not require prior knowledge of the number of modes, as long as the model has a sufficiently large value of k relative to the number of modalities. In contrast, models like GAM and NAM are unable to accurately represent multimodal relationships since they only provide a single estimated outcome per representation. In Section 3.2, we provide a demonstration of this property and also compare the representation of multimodal relationships between MNAM and NAM 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 parameters for k Gaussian distributions and a latent variable for predicting the probability of the mixture of k Gaussian distribution models $(\mathcal{N}_{j,1}(\mu_{j,1},\, \sigma^2_{j,1}), ..., \mathcal{N}_{j,k}(\mu_{j,k},\, \sigma^2_{j,k}), Z^{\pi}_j)$. The left index j is a reference to one of the input features and the right index of the Gaussian distributions is a reference to the number of components for the mixture. 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.

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

The linear property is crucial for the interpretability of additive models. In the case of GAM and NAM, where the output is a linear combination of non-linearly transformed features, the models are interpretable because one can observe how changes in a feature impact the outcome. Therefore, the mixture of Gaussians as the outcome has practical advantages. By leveraging the linear property of a mixture of Gaussians, MNAM can capture the magnitude of changes in the overall mean and uncertainty of predictions in response to variations 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 of how MNAM computes predictions is summarized in Algorithm 1 and the comparison of the architecture for NAM and MNAM is illustrated in Figure 2.

Algorithm 1 Mixture Neural Additive Models

```
  Input: Data: $(X_1, ...X_n)$, Number of Features: $n$, Number of Gaussian Distributions: $k$, Neural Networks for Feature Transformation: $(f_1, ..., f_n)$, Neural Network for Probability: $g$
  Output: Mixture of Gaussian Distributions: $\mathcal{N}_1(\mu_1, \sigma_1^2), ..., \mathcal{N}_k(\mu_k, \sigma_k^2), \pi_1, ..., \pi_k$
  for $i = 1$ to $n$ do
    $\mathcal{N}_{i,1}(\mu_{i,1}, \sigma^2_{i,1}), ..., \mathcal{N}_{i,k}(\mu_{i,k}, \sigma^2_{i,k}), Z_i^\pi = f_i(X_i)$
  end for
  for $i = 1$ to $k$ do
    $\mu_i = \sum_{j=1}^{n} \mu_{j,i}$
    $\sigma_i^2 = \sum_{j=1}^{n} \sigma^2_{j,i}$
  end for
  $\pi_1, ..., \pi_k = g(Z_1^\pi, ..., Z_n^\pi)$
```

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 hard-thresholding (HT) and soft-thresholding (ST) algorithms for the given objectives. The HT algorithm trains or updates a single mode or a Gaussian distribution with a minimum loss, while the ST algorithm trains or updates all k modes or Gaussian distributions with weights computed by the likelihood of each mode on the label. Between the two algorithms, we chose the HT algorithm since it is more numerically stable and computationally efficient compared to the ST model; this is demonstrated in the empirical experiment in Appendix 2.3. The detailed description of the HT and ST algorithm is presented next.

2.2.1 Hard-Thresholding (HT) Algorithm

In order to address the two training objectives, the algorithm calculates two separate losses. For the objective of assigning one or more Gaussian distributions to each relationship, the algorithm computes the Gaussian negative log-likelihood (GNLL) loss for each Gaussian distribution of the outcome against a label. Among k losses, only the minimum factor will be used to compute the total loss, which means only weights used

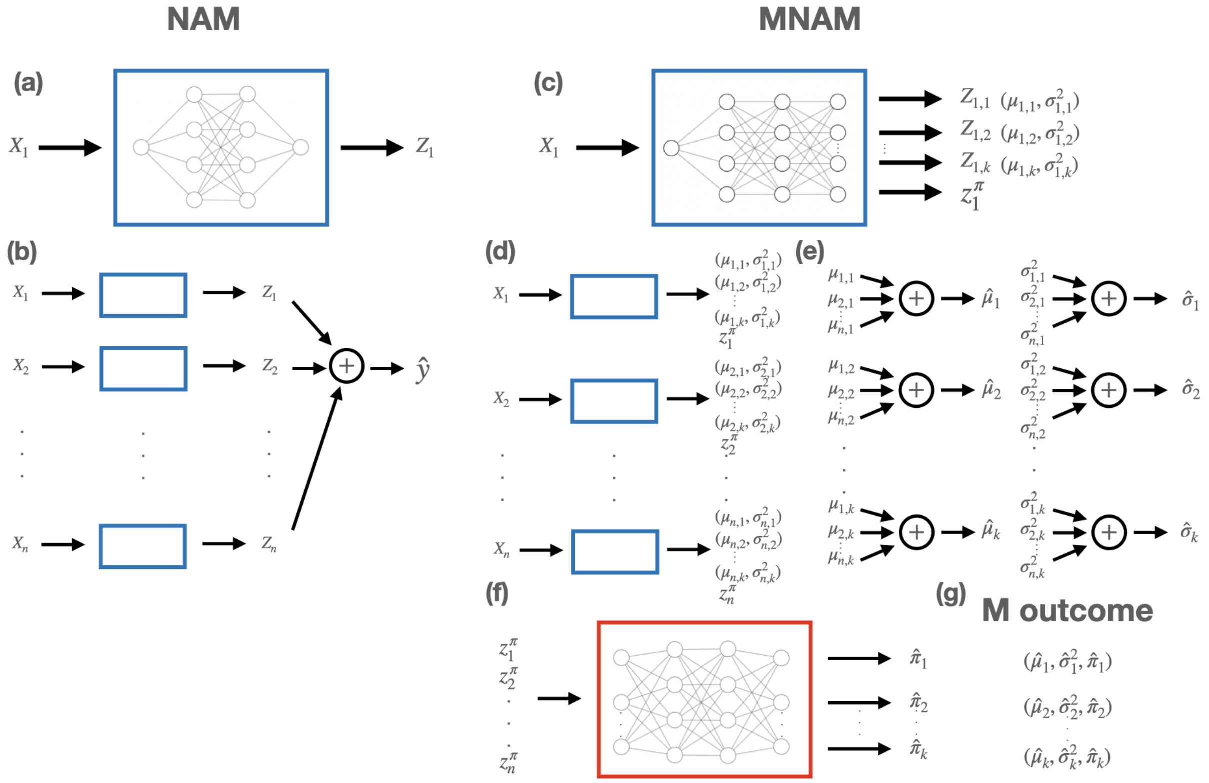

Figure 2: Illustrative schemes of NAM and MNAM network architectures. As shown in (a) and (b), NAM independently maps features into embedding through neural networks and then linearly combines embeddings for a prediction. Similar to NAM, MNAM independently maps features into embeddings through neural 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 $(Z^\pi_1, Z^\pi_2, ..., Z^\pi_n)$ 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.

to compute the minimum loss are updated via the backpropagation. This enables the model to assign one or more Gaussian distributions to learn each relationship. For the second objective, which involves learning to identify subpopulations, the algorithm computes the 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. This loss measures how well MNAM activates the corresponding Gaussian distribution for the input and enables the model to learn to identify subpopulations for a given input to increase the probability of the correct Gaussian distribution for representation. Algorithm 2 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.

2.3 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,

Algorithm 2 Hard-Thresholding (HT) Algorithm

```
**Input:** Data: $(X, Y)$, MNAM: $f$, GNLL loss: $g$, Cross-entropy loss function: $h$, Rate for cross-entropy loss: $\lambda$
$\mathcal{N}_1(\mu_1, \sigma_1), ..., \mathcal{N}_k(\mu_k, \sigma_k), \pi_1, ..., \pi_k = f(X)$
$min\_loss = 0$
**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**
$prob\_loss = h((\pi_1, ..., \pi_k), min\_index)$
$total\_loss = Min\_loss + \lambda \cdot prob\_loss$
```

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

where $P(X, Y|Z = k)$ is the likelihood of kth Gaussian distribution for the given input, 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 3 summarizes the proposed training algorithm.

Algorithm 3 Soft-Thresholding (ST) Algorithm

```
**Input: Data $(X, Y)$, MNAM $f$, GNLL loss $g$, Crossentropy loss function $h$, Rate for cross-entropy loss $\lambda$**
$\mathcal{N}_1(\mu_1, \sigma_1), ..., \mathcal{N}_k(\mu_k, \sigma_k), \pi_1, ..., \pi_k = f(X)$
**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**
$mar\_prob = \sum_{j=1}^{k} gau\_like_j \cdot \pi_j$
$\hat{\pi}_1, ..., \hat{\pi}_k = \frac{gau\_like_1 \cdot \pi_1}{mar\_prob}, ..., \frac{gau\_like_k \cdot \pi_k}{mar\_prob}$
$gau\_loss = \sum_{i=1}^{k} gau\_loss_i \cdot \hat{\pi}_i$
$prob\_loss = h((\pi_1, ..., \pi_k), (\hat{\pi}_1, ..., \hat{\pi}_k))$
$total\_loss = gau\_loss + \lambda \cdot prob\_loss$
```

2.4 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 for California has the task of predicting income and it consists of ten features.

ACS Travel: The ACS Travel dataset for California 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. For both models, we utilized early stopping to reduce overfitting. Optimized parameters from Bayesian optimization can be found in the table from Appendix B. 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.

In comparing deterministic and probabilistic models, we encountered a challenge due to the lack of standardized evaluation metrics. Therefore, we decided to use the mean absolute error (MAE) as a metric for comparison. However, MAE has a limitation. It fails to account for the uncertainty in predictions made by probabilistic models. Even if a probabilistic model accurately predicts a true distribution for the label distribution, it may still receive the same MAE score as a deterministic model if it is correct in predicting the mean of the label distribution. To address this limitation, we transformed NAM into a probabilistic model (pNAM) by setting k=1 in MNAM. We then utilized likelihood as a metric to compare the performance of pNAM with the remaining deterministic models, to emphasize the importance of having multimodal compared to unimodal distribution as an outcome. For the remaining deterministic models, we used the earth mover's distance (EMD) to assess how well they learned to approximate the label distribution. The EMD scores of models can be found in the table from Appendix C. It's worth noting that the EMD score is an unfair evaluation for deterministic models. This is because deterministic models do not learn the uncertainty of the data during training, unlike probabilistic models.

3.1.3 Results

Table 1 displays the MAE scores of NAM and MNAM, as well as the likelihood scores of pNAM and MNAM on datasets described above. MNAM consistently exhibited similar or superior MAE scores compared to NAM across all six datasets. Moreover, MNAM showcased a significantly improved performance in terms of likelihood scores when compared to NAM for all datasets, except for the FICO dataset. Notably, the optimized number of Gaussian distributions for MNAM was 1, which means that pNAM and MNAM are identical models. This finding underscores MNAM's remarkable ability to effectively learn the output distribution, surpassing both NAM and pNAM in this aspect.

Differences in performance between MNAM and NAM differ greatly by datasets. Specifically, the discrepancy in likelihood scores between NAM and MNAM is much more pronounced for the CA Housing dataset compared to the ACS Income dataset. Several explanations could account for this observation. Firstly, the CA Housing dataset might exhibit more intricate interaction relationships among its features, rendering it more challenging for NAM to accurately capture the underlying patterns without any interaction term learning. Conversely, MNAM, with its enhanced capability to model complex interactions, would demonstrate an improved likelihood score on such datasets. Secondly, the CA Housing dataset might possess modes that differ more significantly from one another, making it harder to fit using a single Gaussian distribution for NAM. In this scenario, MNAM would enhance the likelihood score by accommodating the complexity of interaction relationships and the differences among modes within the data.

	NAM		**pNAM**		**MNAM**	
Dataset	MAE↓	Likelihood↑	MAE↓	Likelihood↑	MAE↓	Likelihood↑
CA Housing	$0.48 \pm 9e^{-05}$	NA	NA	$0.58 \pm 6e^{-04}$	$0.46 \pm 4e^{-05}$	0.73 ± 0.001
FICO	2.7 ± 0.002	NA	NA	$0.084 \pm 2e^{-06}$	2.7 ± 0.002	$0.084 \pm 2e^{-06}$
MIMIC	1.5 ± 0.0002	NA	NA	$0.15 \pm 3e^{-06}$	1.5 ± 0.0003	$0.25 \pm 6e^{-05}$
BIKE	3.4 ± 0.0005	NA	NA	$0.069 \pm 3e^{-08}$	3.4 ± 0.0006	$0.092 \pm 1e^{-06}$
ACS Income	37.2 ± 0.003	NA	NA	$0.011 \pm 4e^{-07}$	35.7 ± 0.02	$0.013 \pm 4e^{-07}$
ACS Travel	15.6 ± 0.0004	NA	NA	$0.017 \pm 2e^{-08}$	15.5 ± 0.002	$0.036 \pm 2e^{-05}$

Table 1: MAE score for NAM and MNAM and likelihood score for pNAM and MNAM on CA Housing, FICO, MIMIC, BIKE, ACS Income, and ACS Travel datasets

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 Appendix D. 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.

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

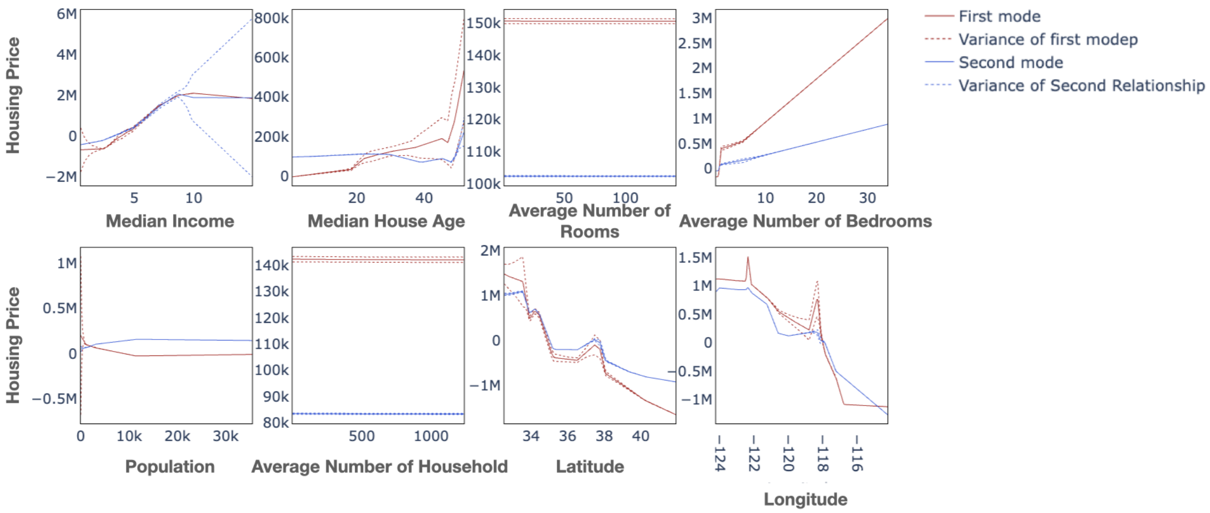

Figure 3: Two relationships between features and label learned by MNAM on CA Housing dataset. Solid lines represent the mean of relationships and dotted lines represent the uncertainties of relationships.

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.

The strength of MNAM in identifying unfairness is more evident when trained on the MIMIC dataset. In Figure 5, the variance or the discrimination of the length of stay among different ethnicities significantly differ between the two modes recognized by MNAM (The left graph of Figure 5). The first relationship, which is a red line, has more variance or discrimination among ethnicities compared to the second relationship, which is a blue line, in the length of stay. If we group the algorithm's output by admission type (the middle graph represents common admission and the right graph represents urgent admission), we recognize the model activates more on relationships with less discrimination among ethnicities with urgent admission and vice versa with common admission. In other words, there is more variance and discrimination in the length of stay among ethnicities for common admission compared to urgent admission. Overall, these findings highlight the superior capabilities of MNAM compared to NAM in identifying the unfairness of a model as NAM will provide the same relationship for all subpopulations.

3.1.5 Training Efficiency

Table 2 shows the comparison of average training time, training time per epoch, and the number of epochs required to train NAM and MNAM on different datasets. As expected, MNAM takes longer to train per epoch than NAM because it has an additional neural network for computing mode weights. However, interestingly, MNAM was faster than NAM in training for half of the datasets, and MNAM required fewer epochs than NAM for all datasets except one. Our hypothesis is that MNAM's assignment of modes to subpopulations effectively shrinks the space for the model to explore, resulting in fewer epochs needed for training, as each mode only needs to represent one subpopulation. In contrast, NAM has only one outcome that must represent multiple subpopulations, causing it to oscillate among subpopulations for representation during training. Overall, MNAM may be more efficient than NAM in terms of training time, despite having more parameters to compute.

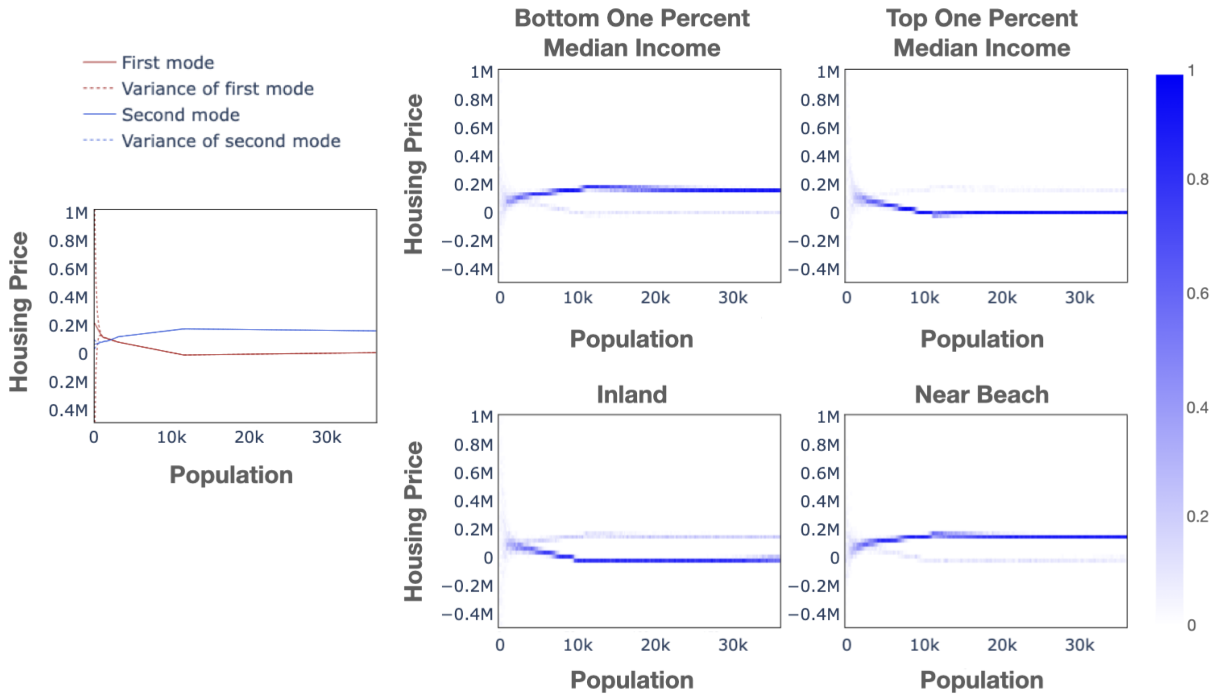

Figure 4: Line graph and heatmaps on the relationship between housing price and population for CA Housing. The first column, a line graph, represents two modes recognized by MNAM between housing prices and populations. Second and third columns, heatmaps, represents changes in the activation of two modes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

	Training Time (Seconds)↓		Training Time per Epoch (Seconds)↓		Number of Epochs↓	
Dataset	NAM	MNAM	NAM	MNAM	NAM	MNAM
CA Housing	636.38	411.73	0.72	0.84	885.36	489.85
FICO	326.26	337.54	0.78	0.85	417.34	397.67
MIMIC	173.10	278.58	0.93	1.09	186.64	256.53
BIKE	410.42	477.62	1.37	1.85	298.76	258.36
ACS Income	1460.31	896.41	3.22	4.47	453.65	200.40
ACS Travel	2411.51	946.55	3.86	5.75	624.43	164.58

Table 2: Comparisons of average training time in seconds, training time per epoch in seconds, and number of epochs between MNAM and NAM

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

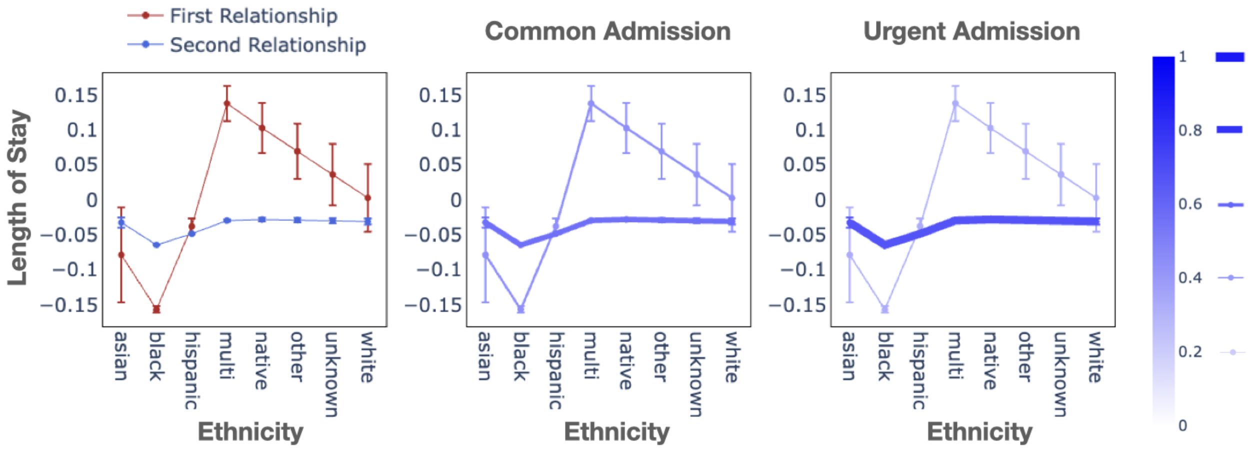

Figure 5: Line graphs of the relationship between the length of stay and the ethnicity for MIMIC with whisker representing a variance. The left graph represents two modes recognized by MNAM between the length of stay and the ethnicity. The middle and right graphs represent changes in the activation of two modes from changes in admission type. Except for admission type, all other remaining features have been fixed to mean values. The magnitude of the mode's activation is illustrated through the intensity of color and thickness of lines. The darker blue and thicker line represents higher activation of a mode. The blue color bar and different thickness of lines on the right side of the color bar represents the magnitude of a mode's activation.

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 for NAM when variables that differentiate subpopulations are latent variables, which is the case for $X_2 = 2$ and $X_2 = 3$ in the synthetic dataset. NAM tries to represent multiple relationships with one relationship, as shown in the second column of Figure 6. 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 6.

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 3 shows the MAE 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

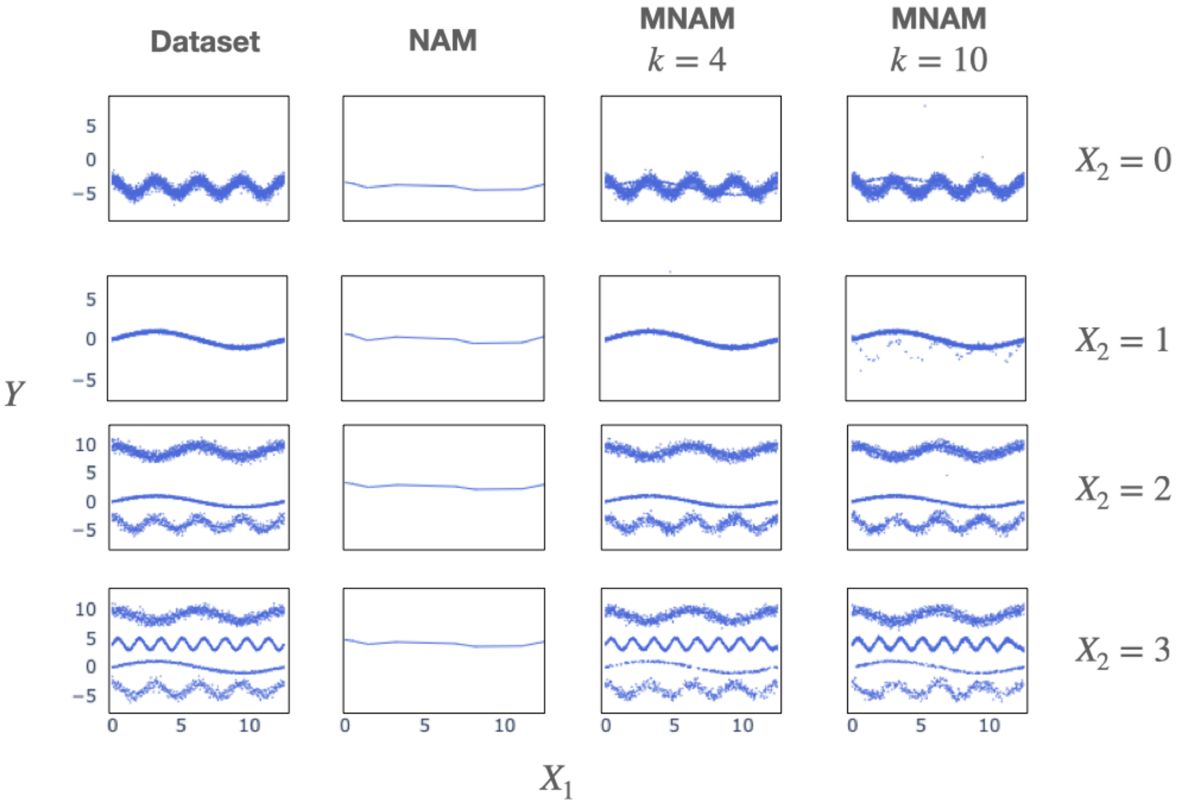

Figure 6: NAM versus MNAM on a dataset that has variables that identify subpopulations as observed and latent variables. The left column is a scatter plot for a dataset with different values of X_2. The remaining columns represent predictions from training on the dataset for NAM, MNAM with $k = 4$, and 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.

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 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.

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. For transforming each feature, various functions have been used such as boosted decision trees Nori et al. (2019) and piecewise linear functions Zuur (2012). 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.

	Direct Input and Output Relationships			Complex Input and Output Relationships	
Datasets	LR	NAM	MNAM	EBM	GBT
CA Housing	0.54	0.48	0.46	0.34	0.31
FICO	3.38	2.7	2.7	2.5	2.4
MIMIC	1.5	1.5	1.5	1.5	1.5
BIKE	3.65	3.4	3.4	3.4	3.4
ACS Income	40.0	37.2	35.7	33.3	31.8
ACS Travel	16.8	15.6	15.5	14.2	13.8
Complexity					→
Interpretability	←				

Table 3: MAE 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.

Furthermore, it is more flexible for interpretation as it is 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. The model requires users to read two graphs for interpretation. One is for a line graph on the relationship between label and feature of interest and another one is for a heatmap on interaction terms. Users have to mentally visualize the relationship by looking at two graphs to understand how the relationship changes from a change in other features. Compared to GA2M, in the proposed MNAM users have to only look at one graph and don't have to mentally visualize as the relationship is fixed for MNAM.

Mixture Density Networks (MDNs) 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. MDNs is not a form of a Generalized Additive Model but more of a DNN with a mixture of k Gaussian distributions as an outcome. For DNN and MDN, the relationship between a feature of interest and a label will completely change from changes in other features. It is difficult to compare all possible relationships and describe how they differ from each other. Thus, it is difficult to show quantitatively how a relationship changes from changes in other features. Meanwhile, MNAM has k-fixed relationships, so it is easier to compare and describe differences among relationships especially when k is small. Unlike DNN and MDN, MNAM is able to show quantitively how a relationship changes by showing how much intensity of activation on one relationship increases or decreases in exact percentage-wise from changes in other features.

Furthermore, MNAM can show the relationship for minority subpopulations in the case when the subpopulation is differentiated by latent variables by assigning one of the k relationships to the subpopulation. For deterministic models like NAM and DNN, they will fail as they are limited to showing only one relationship. This is demonstrated in Section 3.2 as the synthetic dataset for pedagogical experiment contains multiple subpopulations differentiated by latent variables. This can be critical, for example, in medical applications, where a drug might be effective in a certain subgroup of the population, tools like MNAM, would allow identifying from data modes or outliers that might not fit the general expected therapeutic trend.

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.

MNAM learns means and variances of the outcome by minimizing the GNLL. Yet this method has been known to have limitations as it could fail to learn the optimal means and variances of the outcome Stirn et al. (2023). For our future works, we will utilize a Bayesian neural network to overcome this limitation. Within the Bayesian neural network, there are different methods for learning the posterior distribution of weights, which could be classified into variational inference Gal & Ghahramani (2016); Ovadia et al. (2019) and Markov Chain Monte Carlo (MCMC) methods Welling & Teh (2011); Zhang et al. (2019). Both methods have different trade-offs. Variational inference is less computationally expensive but lacks expressive power and vice versa for MCMC. We will explore both methods to find an optimal balance between trade-offs.

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.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/gal16.html`.

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.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf`.

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.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5593–5613. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/stirn23a.html`.

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`.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

Alain F Zuur. *A beginner's guide to generalized additive models with R*. Highland Statistics Limited Newburgh, 2012.

A 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 4 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. Regarding 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 4: Comparision of HT algorithm and ST algorithm on data from pedagogic study

B Table of optimized parameters for MNAM

	Learning	Weight	Output	Number of	Cross-entropy
Dataset	rate	decay	penalty	Gaussian distribution	loss
CA Housing	0.009896	3.8512e-05	0.03363	2	0.6118
FICO	0.05	1e-06	0.0166	1	0.7762
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 5: Optimized parameters for MNAM on six datasets

C Table of Earth Mover's Distance score for models

	Direct Input and Output Relationships			Complex Input and Output Relationships	
Datasets	LR	NAM	MNAM	EBM	GBT
CA Housing	0.29	0.24	0.077	0.11	0.09
FICO	1.16	0.73	0.60	0.51	0.51
MIMIC	1.36	1.43	0.24	1.32	1.25
BIKE	3.06	2.50	0.26	2.45	2.43
ACS Income	27.1	21.3	7.4	14.5	12.9
ACS Travel	14.2	12.8	3.1	8.9	8.7
Complexity	→				
Interpretability	←				

Table 6: 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.

D Relationships plots on other datasets

D.1 FICO

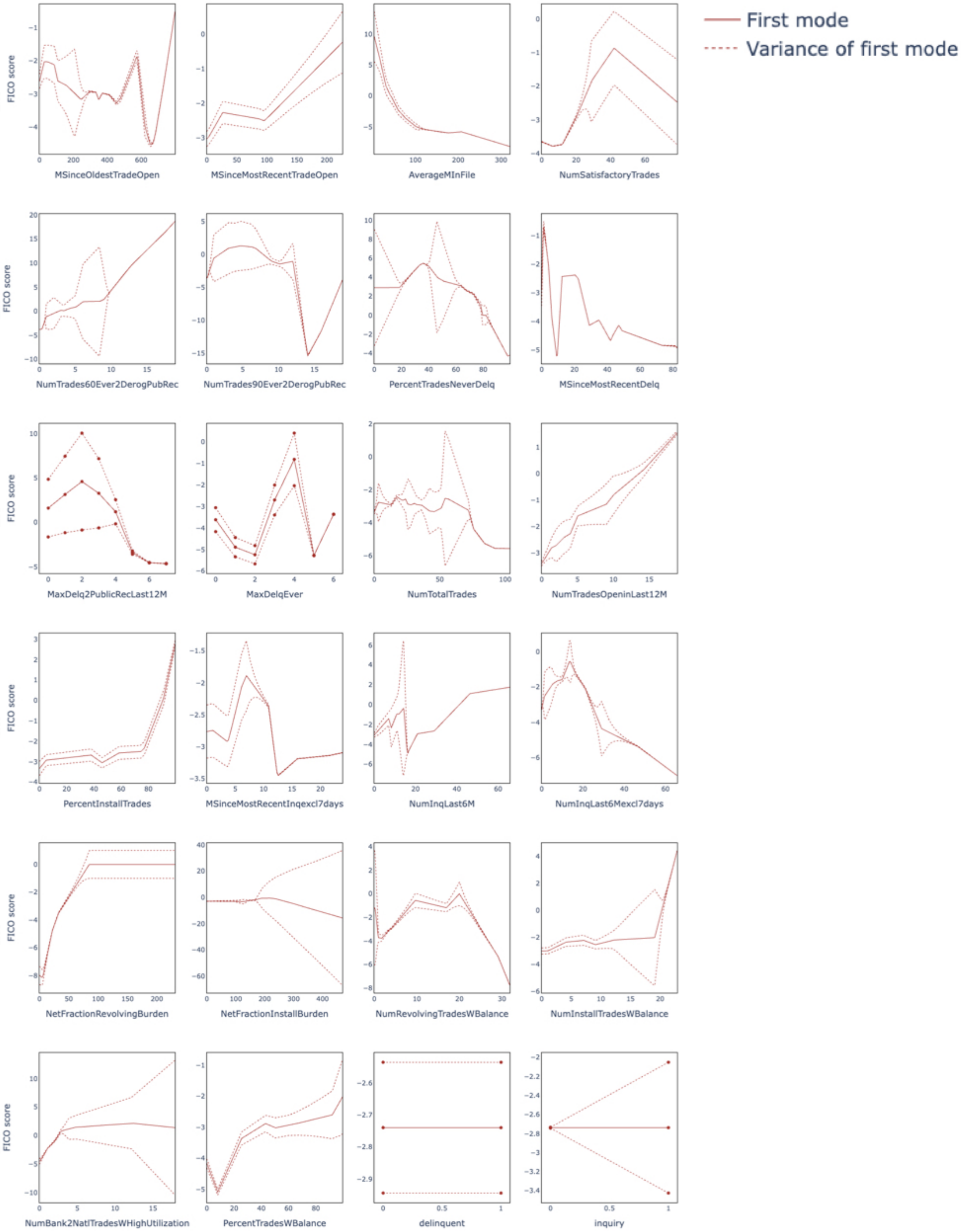

Figure 7: Learned relationships between features and labels for the MNAM on FICO datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

D.2 MIMIC

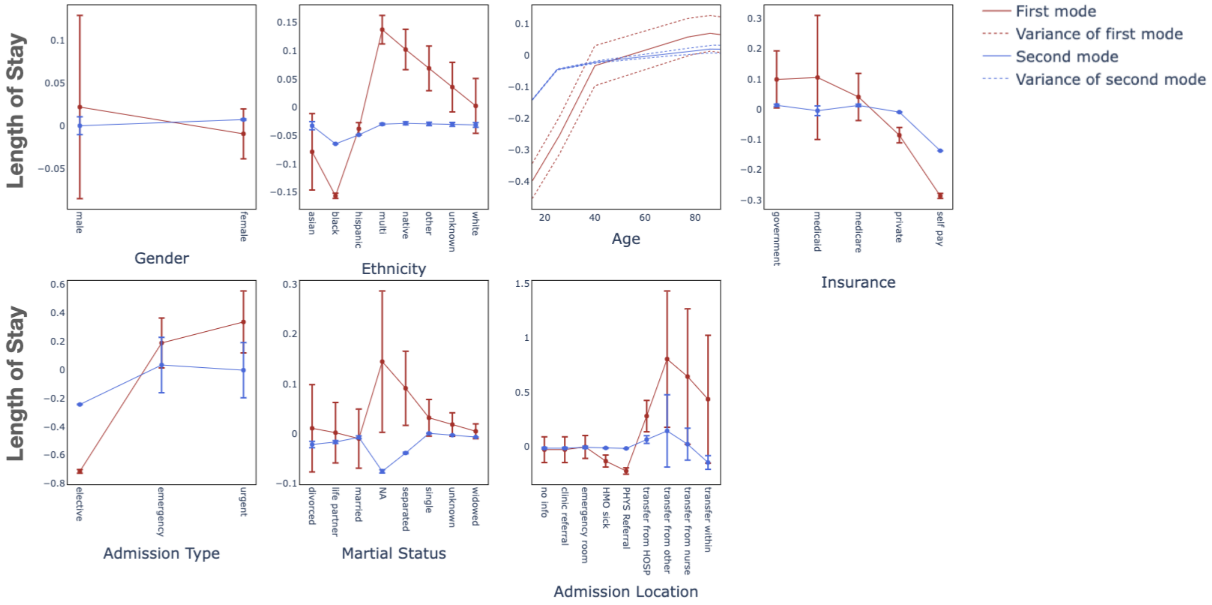

Figure 8: Learned relationships between features and labels for the MNAM on MIMIC datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

D.3 BIKE

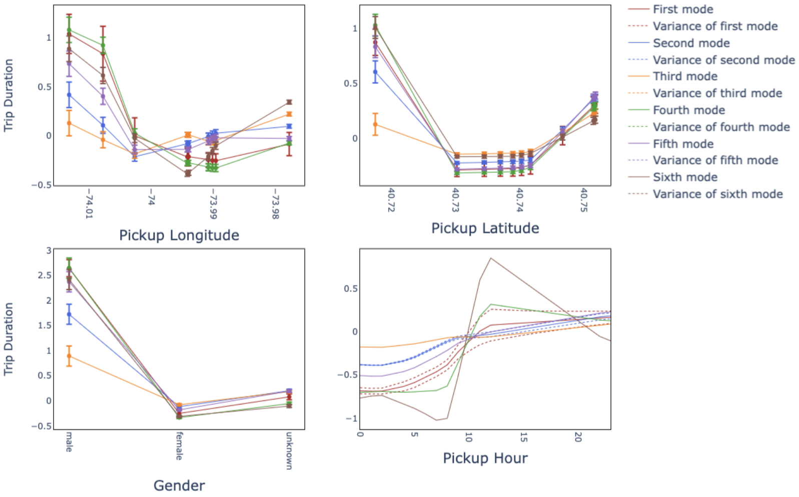

Figure 9: Learned relationships between features and labels for the MNAM on BIKE datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

D.4 ACS Income

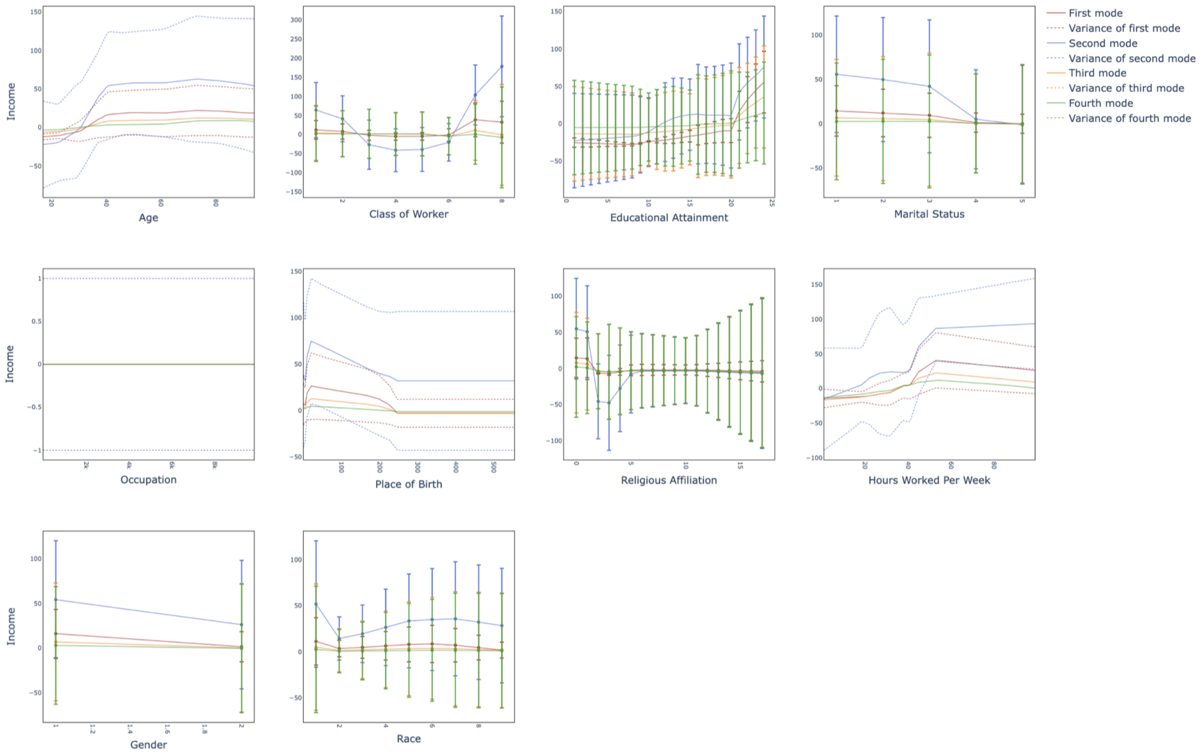

Figure 10: Learned relationships between features and labels for the MNAM on ACS Income datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.

D.5 ACS Travel

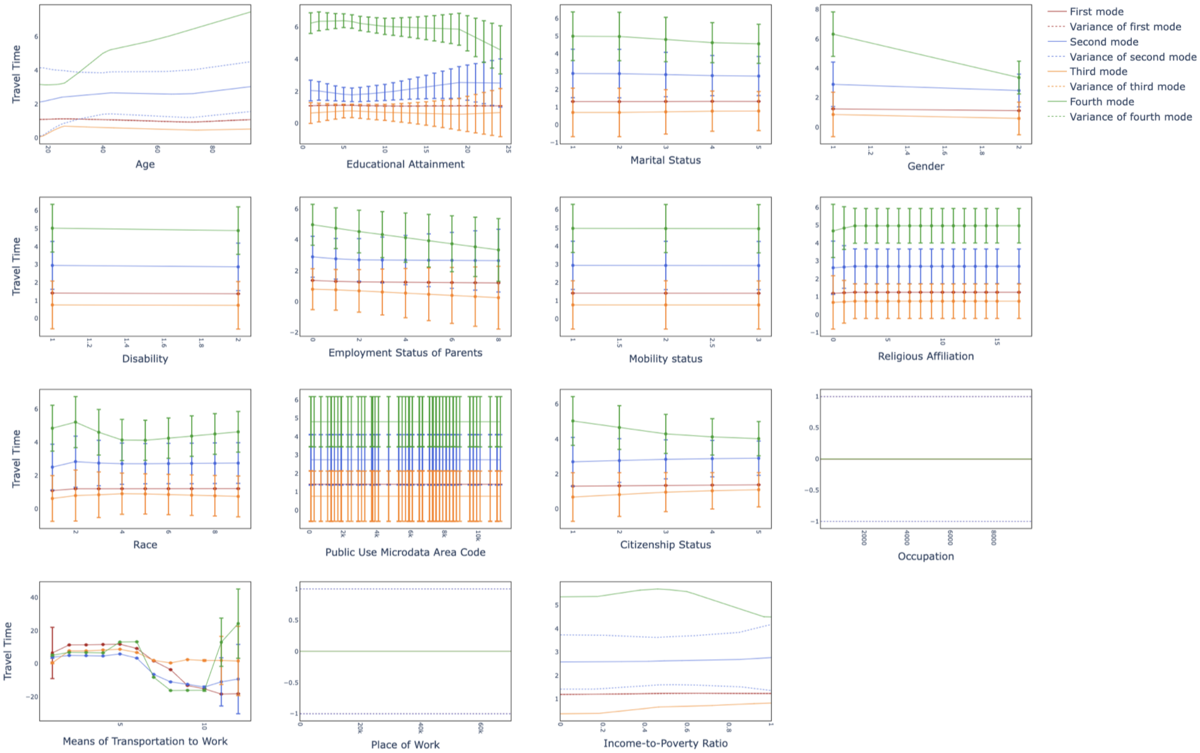

Figure 11: Learned relationships between features and labels for the MNAM on ACS Travel datasets. Solid lines represent the mean of the relationships and dotted lines represent their uncertainties.