A Generalized Additive Models: Extended Related Work

As with many families of machine learning algorithms, the differences among GAM algorithms lie in (a) the functional form of the shape functions f_i, (b) the learning algorithm used for their estimation and (c) regularity assumptions and regularization. Two important properties that all GAMs share are (1) the ability to learn non-linear transformations for each feature and (2) additively combining these shape functions (prior to applying the link function) to create modularity that aids interpretability by allowing users to examine shape functions one-at-a-time.

Typically, GAMs have relied on splines and backfitting algorithms for estimation (Hastie and Tibshirani, 1987), with subsequent works focusing on improving efficiency and stability through penalized regression splines (Wood, 2003) and fast, stable fitting algorithms (Wood, 2001). Spline-based GAMs are typically fitted using the backfitting algorithm, an iterative procedure that starts with initial estimates of the smooth functions for each predictor variable. The algorithm then repeatedly updates each function by fitting a weighted additive model to the residuals of the other functions until convergence is achieved. The weights are determined by the current estimates of the other functions and the link function in the case of generalized additive models.

Modern approaches leverage machine learning advances. Explainable Boosting Machines (EBMs) (Lou et al., 2012; 2013; Caruana et al., 2015) model the shape functions using decision trees, which are fitted using a variant of gradient boosting called cyclic gradient boosting. The model iteratively learns the contribution of each feature and interaction term in a round-robin fashion, using a low learning rate to ensure that the order of features does not affect the final model. This cyclic training procedure helps mitigate the effects of colinearity among predictors by providing opportunity for data-driven credit attribution among the features while preventing multiple counting of evidence. EBMs are also popular because they can accurately capture steps in the shape functions, which is important for modeling discontinuities in data, such as treatment effects in medical data.

More recently, Neural Additive Models (NAMs) (Agarwal et al., 2021) and follow up works (Chang et al., 2021; Dubey et al., 2022; Radenovic et al., 2022; Xu et al., 2022; Enouen and Liu, 2022; Bouchiat et al., 2024) use multilayer perceptrons (MLPs), as non-linear transformations, to model the shape functions f_i. As a result, NAMs can be optimized using variants of gradient descent by leveraging automatic differentiation frameworks.

Finally, GAMs have also found applications in time-series forecasting, with models such as Prophet (Taylor and Letham, 2018) and NeuralProphet (Triebe et al., 2021). Interestingly, the 1-layer versions of the recently proposed Kolmogorov-Arnold Networks (KANs) (Liu et al., 2024) may be viewed as GAMs with spline based shape functions.

B Dataset Details

In this section, we provide details on the datasets used in our empirical evaluations of GAMformer and other baselines in Section 4 of the main paper.

B.1 TabPFN Test Datasets

As test dataset, we used the 30 datasets used in Hollmann et al. (2023) which were obtained from OpenML (Vanschoren et al., 2014). These were chosen because they contain up to 2000 samples, 100 features and 10 classes, show in Table 1.

B.2 Binary Classification

Churn dataset. The Telco Customer Churn Dataset is a binary classification dataset for predicting potential subscription churners in a telecom company, containing customer information and churn-related features.

Adult dataset. The Adult dataset Dua and Graff (2017), also known as the "Census Income" dataset, is a widely-used benchmark for binary classification, predicting whether an individual's annual income exceeds $50,000 based on 14 attributes from the 1994 United States Census Bureau data.

Table 1: Test dataset names and properties, taken from Hollmann et al. (2023). Here *did* is the OpenML Dataset ID, *d* the number of features, *n* the number of instances, and *k* the number of classes in each dataset.

did	name	d	n	k
11	balance-scale	5	625	3
14	mfeat-fourier	77	2000	10
15	breast-w	10	699	2
16	mfeat-karhunen	65	2000	10
18	mfeat-morphological	7	2000	10
22	mfeat-zernike	48	2000	10
23	cmc	10	1473	3
29	credit-approval	16	690	2
31	credit-g	21	1000	2
37	diabetes	9	768	2
50	tic-tac-toe	10	958	2
54	vehicle	19	846	4
188	eucalyptus	20	736	5
458	analcatdata_authorship	71	841	4
469	analcatdata_dmft	5	797	6

did	name	d	n	k
1049	pc4	38	1458	2
1050	pc3	38	1563	2
1063	kc2	22	522	2
1068	pc1	22	1109	2
1462	banknote-authentication	5	1372	2
1464	blood-transfusion-...	5	748	2
1480	ilpd	11	583	2
1494	qsar-biodeg	42	1055	2
1510	wdbc	31	569	2
6332	cylinder-bands	40	540	2
23381	dresses-sales	13	500	2
40966	MiceProtein	82	1080	8
40975	car	7	1728	4
40982	steel-plates-fault	28	1941	7
40994	climate-model-...	21	540	2

Table 2: Comparison of GAMformer with other GAM variants and full complexity models on various datasets. We report ROC-AUC (%) (higher is better) and the standard error over 10 fold cross-validation. We also report results by pyGAM (Servén and Brummitt, 2018).

	GAMs				*Full Complexity*		
	GAMformer (ours)	EBM (Main effects)	Logistic Regression	pyGAM (Main effects)	EBM	XGBoost	Random Forest
Churn	81.69 $\pm$ 0.1	83.59 $\pm$ 0.1	81.66 $\pm$ 0.1	82.03 $\pm$ 0.0	83.68 $\pm$ 0.1	83.53 $\pm$ 0.0	82.07 $\pm$ 0.0
Support2	80.84 $\pm$ 0.1	82.36 $\pm$ 0.0	81.1 $\pm$ 0.0	81.74 $\pm$ 0.2	83.51 $\pm$ 0.0	84.03 $\pm$ 0.0	83.93 $\pm$ 0.0
Adult	90.05 $\pm$ 0.0	93.05 $\pm$ 0.0	90.73 $\pm$ 0.0	91.55 $\pm$ 0.0	93.07 $\pm$ 0.0	93.16 $\pm$ 0.0	91.8 $\pm$ 0.0
MIMIC-2	82.22 $\pm$ 0.0	85.15 $\pm$ 0.0	81.62 $\pm$ 0.0	83.89 $\pm$ 0.1	86.36 $\pm$ 0.1	87.29 $\pm$ 0.0	87.31 $\pm$ 0.0
MIMIC-3	74.41 $\pm$ 0.1	81.14 $\pm$ 0.0	78.05 $\pm$ 0.0	79.95 $\pm$ 0.1	82.52 $\pm$ 0.1	83.32 $\pm$ 0.0	81.28 $\pm$ 0.1

MIMIC-II dataset. The MIMIC-II dataset Lee et al. (2011b) is a publicly-available database of clinical data from diverse ICU patients, integrating demographics, vital signs, lab results, medications, procedures, notes, and imaging reports, along with mortality outcomes.

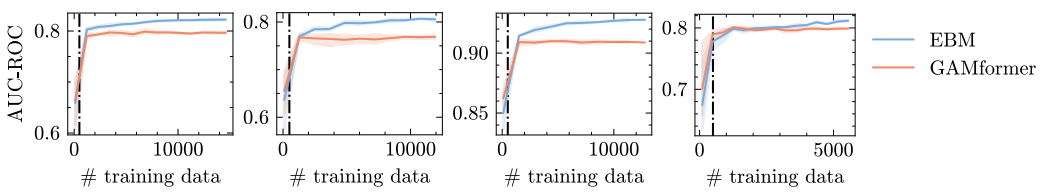
MIMIC-III dataset. The MIMIC-III dataset Johnson et al. (2016) expands on MIMIC-II, with a larger patient cohort, more recent records, enhanced data granularity, and the inclusion of free-text imaging report interpretations.

SUPPORT2 dataset. The SUPPORT2 dataset Connors Jr et al. (1996) contains medical information from critically ill hospitalized adults, compiled to study the relationships between medical decision-making, patient preferences, and treatment outcomes, with variables spanning demographics, physiology, diagnostics, treatments, and survival/quality of life outcomes.

C Properties of GAMformer

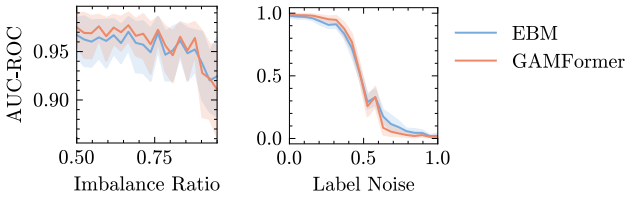
C.1 Data Scaling

To assess GAMformer's ability to generalize to datasets containing more datapoints than it saw during training, i.e. larger context sizes, we conducted an experiment that varied the number of training data points and evaluated the impact on ROC-AUC performance using a consistent validation split. To ensure the robustness of our findings, we sampled training datasets three times with replacement for each training size. The results in Figure 8 demonstrate that GAMformer's ROC-AUC improves across datasets when the number of training examples is up to twice the number of training examples seen during training. For comparison, we also evaluated the performance of EBMs under the same



(a) MIMIC-II (b) MIMIC-III (c) Adult (d) Support 2

Figure 8: Demonstration of the ability of GAMformer to scale beyond the datapoints seen during training while leveraging the additional data points to increase its performance. The dashed vertical line denotes the number of in-context examples seen during training (500).



(a) Imbalanced Data (b) Noisy Labels

Figure 9: Comparison of GAMformer and EBMs in terms of (a) performance on class imbalanced data and (b) robustness to noisy labels. The shaded areas represent the 5% and 95% confidence intervals estimated using 1000 bootstrap samples.

conditions. While EBMs also exhibited improvements in ROC-AUC with increased training data, they achieved higher accuracy when provided with a larger number of examples. This observation highlights a limitation of GAMformer in its ability to fully leverage additional training samples.

C.2 Class Imbalance

To compare GAMformer's sensitivity to class imbalance with that of EBMs, we conduct the following analysis. First, we sample 300 data points from two centroids in a 20-dimensional feature space, creating a binary classification problem. We then vary the ratio of the two classes to introduce increasing levels of imbalance in the sampled data. Next, we split the data into train and test sets using a 75% to 25% split and evaluate the performance using the AUC-ROC metric. We repeat the experiment 10 times for each data ratio. Our results are shown in Figure 9a, the shaded area are the 5%, 95% confidence intervals estimated using 1000 bootstrap samples. We see that GAMformer performs on average better than EBMs in this setting and shows no inherent sensitivity to class imbalance.

C.3 Noise Robustness

To gain a deeper understanding of GAMformers' sensitivity to noisy or incorrect labels, we conducted an experiment similar to the one described in Appendix C.2. We generated 300 data points and randomly perturbed the labels in the train split with increasing probability (75%, 25% train/test split), repeating each experiment 10 times. Figure 9b illustrates our findings. Once again, we observed that GAMformer exhibits a sensitivity to noisy labels comparable to that of EBMs.

D Synthetic Data Priors

We use the same synthetic data generation process proposed in Prior-Data-Fitted Networks (PFNs) (Hollmann et al., 2023; Müller et al., 2022) and provide a brief summary of the process.

TabPFN is trained on two synthetic data priors, which are mixed during training.TabPFN introduced a synthetic data prior based on Structural Causal Models (SCMs). SCMs are particularly suitable for

modeling tabular data as they capture causal relationships between columns, a strong prior in human reasoning. An SCM comprises a set of structural assignments (mechanisms) where each mechanism is defined by a deterministic function and a noise variable, structured within a Directed Acyclic Graph (DAG). The causal relationships are represented by directed edges from causes to effects, facilitating the modeling of complex dependencies within the data. To instantiate a PFN prior based on SCMs, one defines a sampling procedure to create supervised learning tasks. Each dataset is generated from a randomly sampled SCM, including its DAG structure and deterministic functions. Nodes in the causal graph are selected to represent features and targets, and samples are generated by propagating noise variables through the graph. This process results in features and targets that are conditionally dependent through the DAG structure, capturing both forward and backward causation (Hollmann et al., 2023). This allows for the generation of diverse datasets.

The second prior samples of synthetic data using Gaussian Processes (GPs) (Rasmussen and Williams, 2006) with a constant mean function and a radial basis function (RBF) kernel to define the covariance structure. Hyperparameters such as noise level, output scale, and length scale are sampled from predefined distributions to introduce variability. Depending on the configuration, input data points can be sampled uniformly, normally, or as equidistant points and the target column is generated by passing the input data through the GP. This prior gives the model the ability to learn smoother functions.

For multi-class prediction, scalar labels are transformed into discrete class labels by partitioning the scalar values into intervals corresponding to different classes, ensuring the synthetic data is suitable for imbalanced multi-class classification tasks.

Finally, both priors are combined by sampling batches of data from each prior with different probabilities during training. In all of our experiments we sampled from the SCM and GP prior with probability 0.96 and 0.04, respectively.

E Training Details

In GAMformer, we used a transformer model with 12 hidden layers, 512 embedding size and 4 heads per attention. To bin the shape functions and all features we used 64 bins. For training, we use the AdamW (Loshchilov and Hutter, 2019) optimizer ($\beta_1 = 0.9$) and cosine learning rate schedule with initial learning rate of 3e-5, 20 warm up epochs and minimum learning rate of 1e-8 for 25 days on a A100 GPU with 80Gb of memory. We used mixed precision training. Each epoch (arbitrarily) consists of 65536 synthetic datasets; the model trained for 1800 epochs, meaning it saw over 100M synthetic datasets. We used a batch size of 8, that we doubled at epoch 20, 50, 200 and 1000. Each synthetic dataset consisted of 500 samples that were split into training and test portions using using a uniform sampling of the training fraction, and used a number of features drawn uniformly between 1 and 10.

F Higher-order effects

To handle higher-order effects, we compute the best pairs with the FAST algorithm (Lou et al., 2013) and evaluate GAMformer on the top pairs using the following ratios of features:

$$\mathcal{P} = [0.01p, 0.05p, 0.1p, 0.2p, 0.4p, 0.8p, 0.9p]$$

where we recall that p denotes the number of features. We round off each ratio to determine the number of target pair features, evaluate performance on hold-out validation data from the training set, and select the number of pairs with the best validation performance. The model is then fitted on the entire training dataset. This involves doing $|\mathcal{P}| + 1$ forward passes, which is unproblematic as doing one forward pass is very fast, even on a CPU. One could also vectorialize all computations which we do not do given the low fitting time.

G SHAPE FUNCTIONS

In this section, we show complementary results on the shape functions estimates from GAMformer and EBM (main effects only) on the MIMIC-II (Lee et al., 2011a) (complementary to the plots in Figure 7) and on the MIMIC-III datasets.

G.1 MIMIC-II DATASET

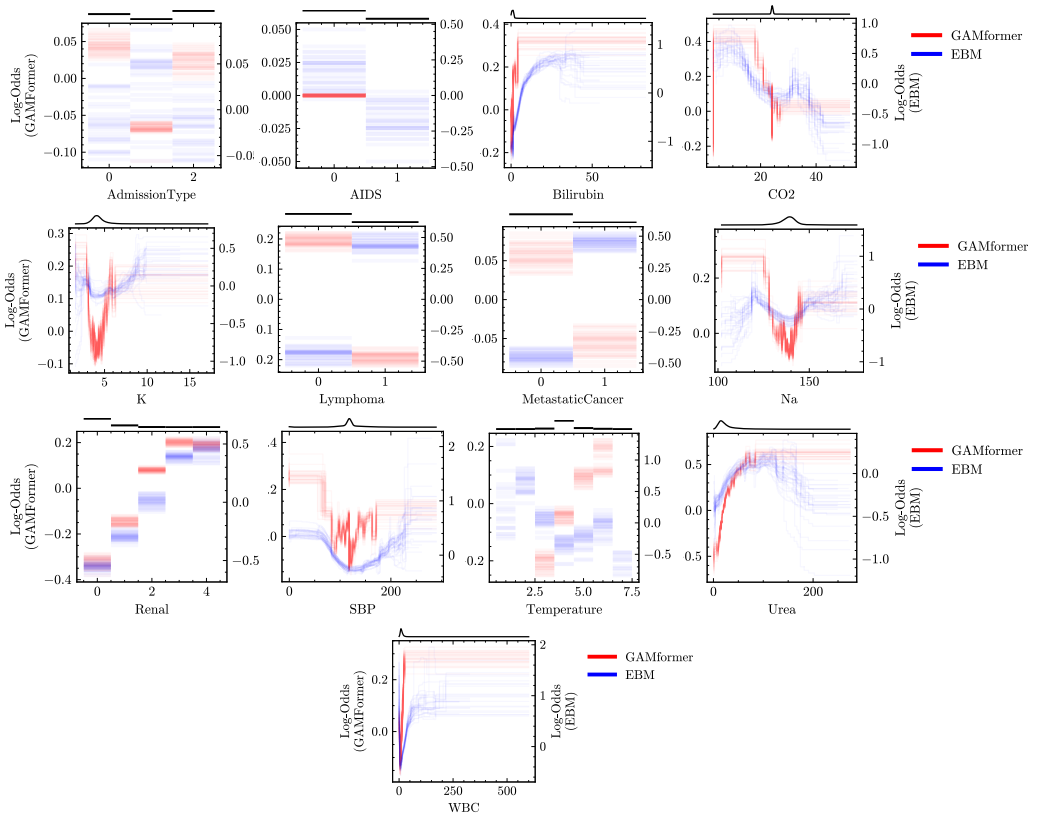


Figure 10: The remaining shape functions derived from GAMformer and EBMs on the **MIMIC-II dataset** for critical clinical variables. The plot above each figure shows the data density. There are interesting differences between the EBM and GAMformer shape plots for several of the categorical variables. Although different GAM algorithms do not usually learn identical functions, we are investigating to better understand these differences.

G.2 MIMIC-III DATASET

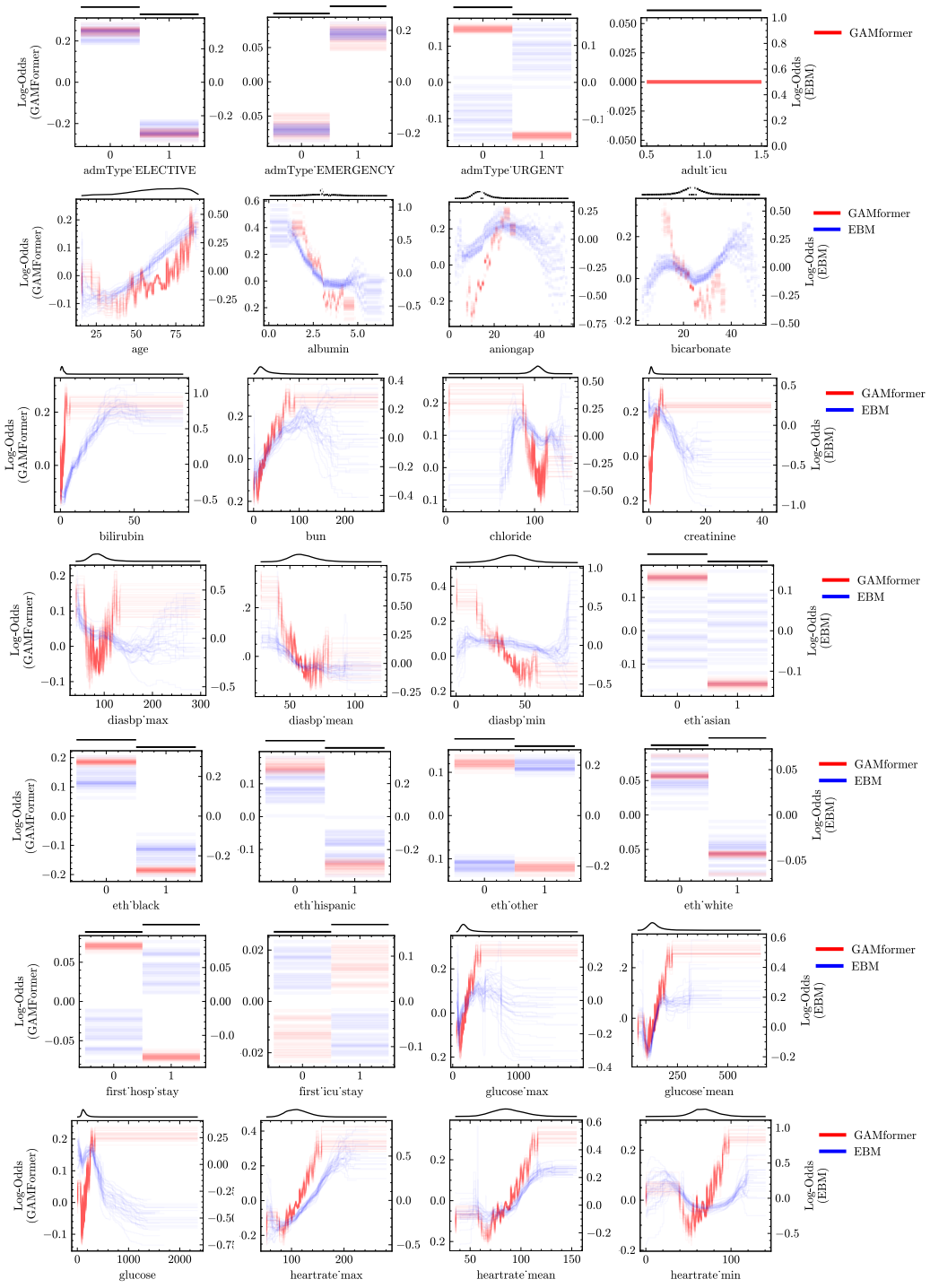


Figure 11: The shape functions derived from GAMformer and EBMs on the **MIMIC-III dataset** for critical clinical variables. The plot above each figure shows the data density. The results are based on 30 models for both GAMformer and EBMs, each fitted on 10,000 randomly selected data points. There are interesting differences between the EBM and GAMformer shape plots for several of the categorical variables. Although different GAM algorithms do not usually learn identical functions, we are investigating to better understand these differences.

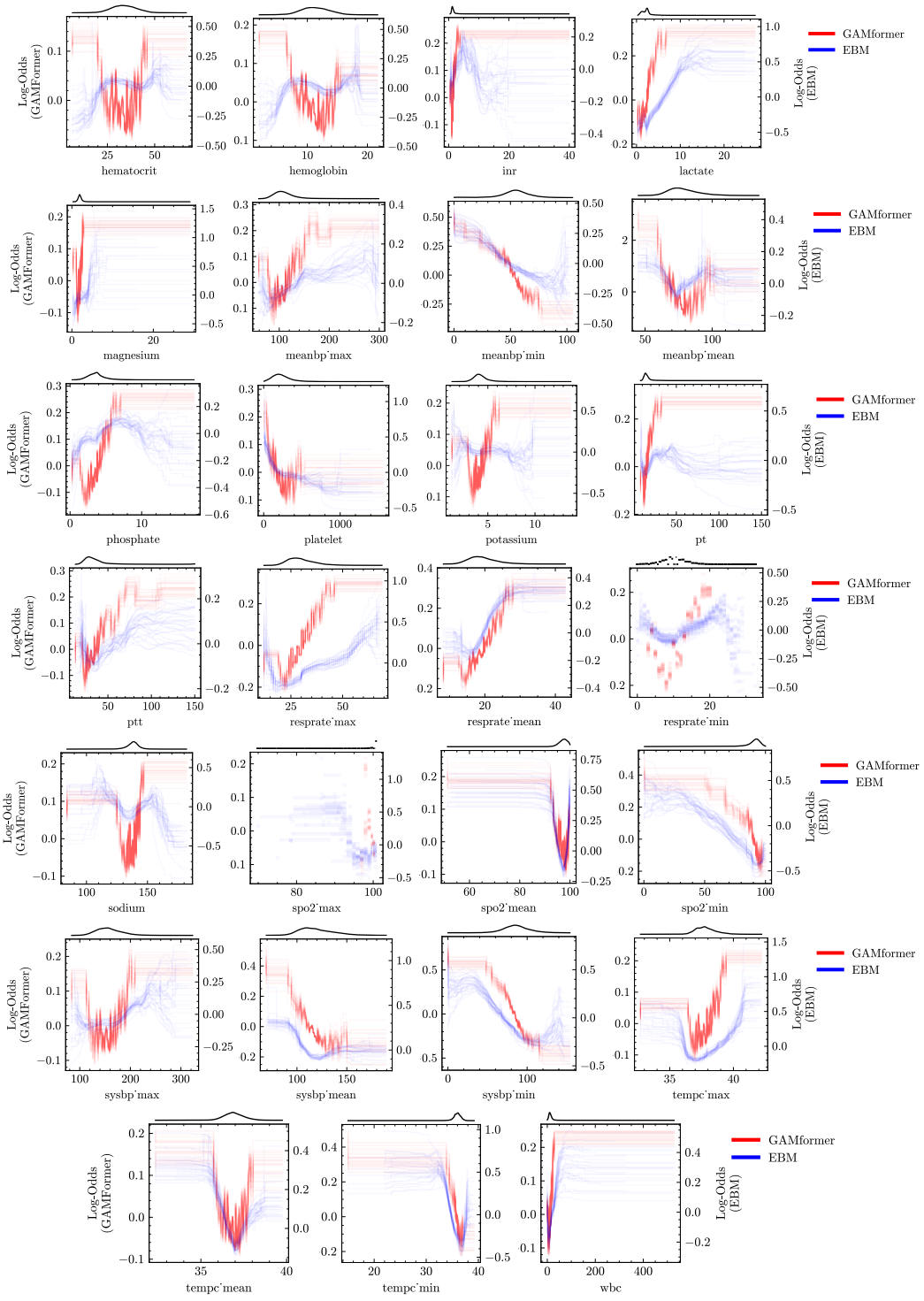


Figure 12: The remaining shape functions derived from GAMformer and EBMs applied to the **MIMIC-III dataset** for critical clinical variables. The plot above each figure shows the data density in the training set. The results are based on 30 models for both GAMformer and EBMs, each fitted on 10,000 randomly selected data points.