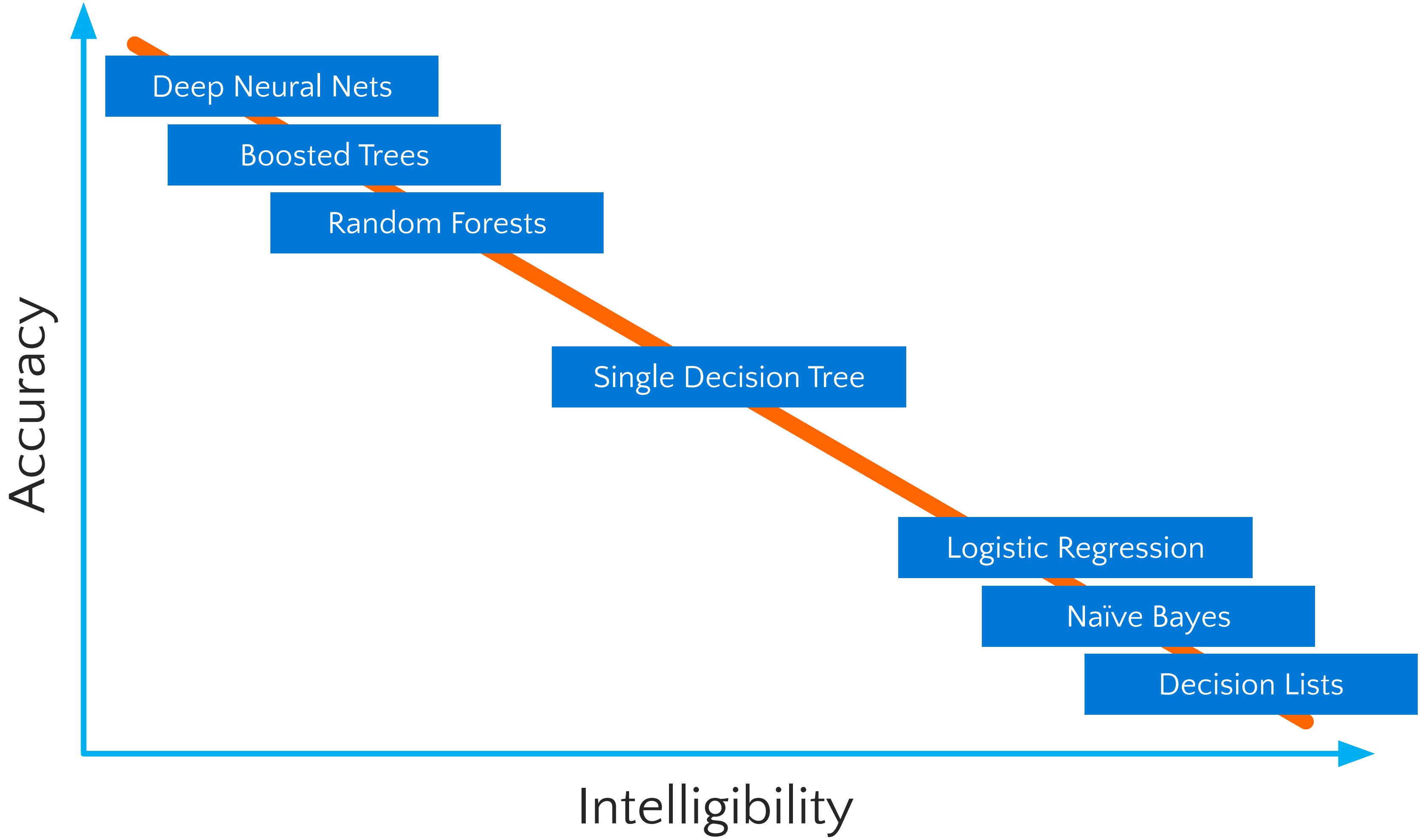
NAMs: Neural Additive Models
Interpretable Machine Learning With Neural Nets

Rishabh Agarwal, Levi Melnick, Ben Lengerich, Nicholas Frosst, Xuezhou Zhang, Rich Caruana, Geoffrey Hinton

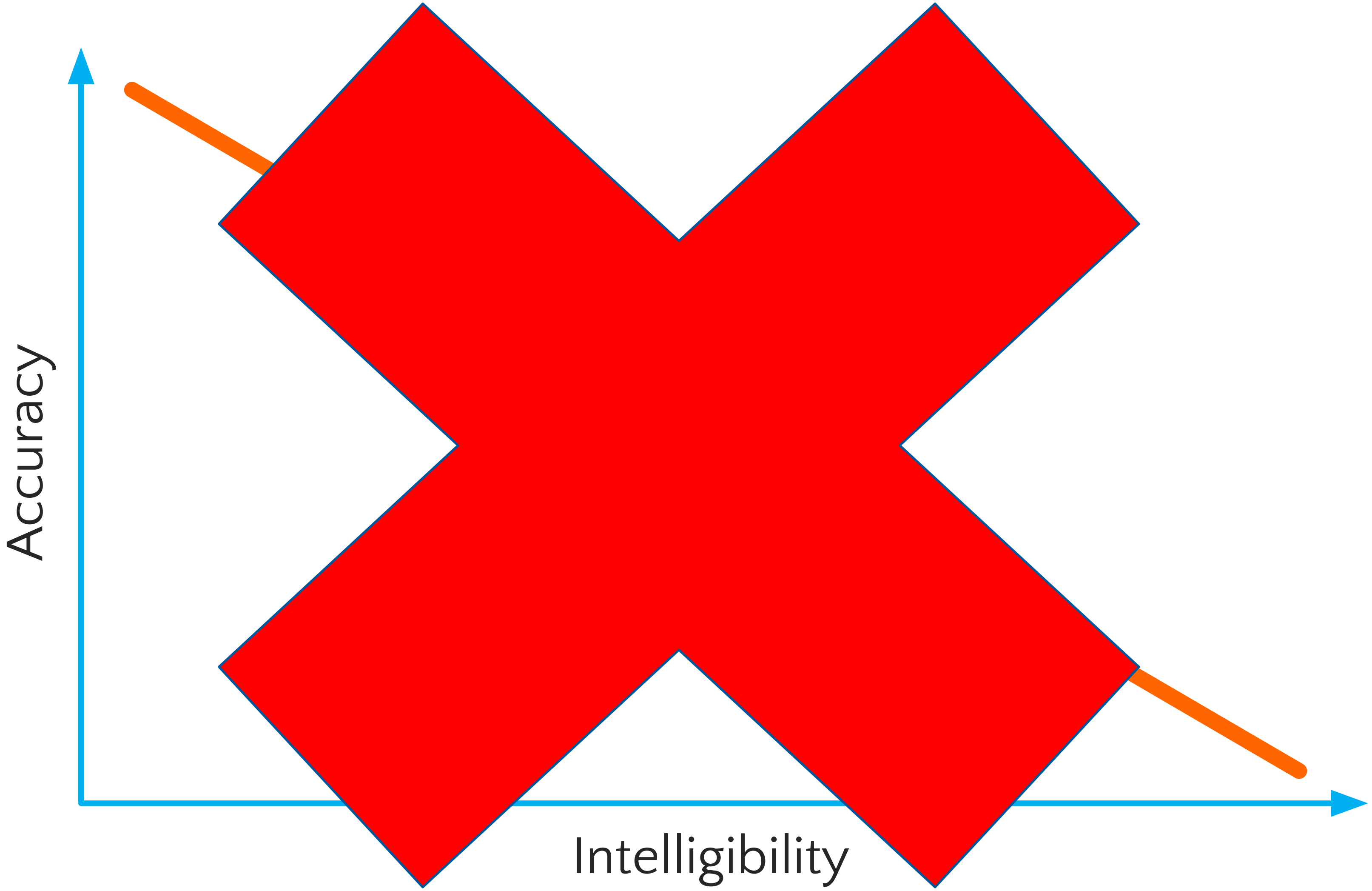
Introduction to GAMs

Generalized Additive Models

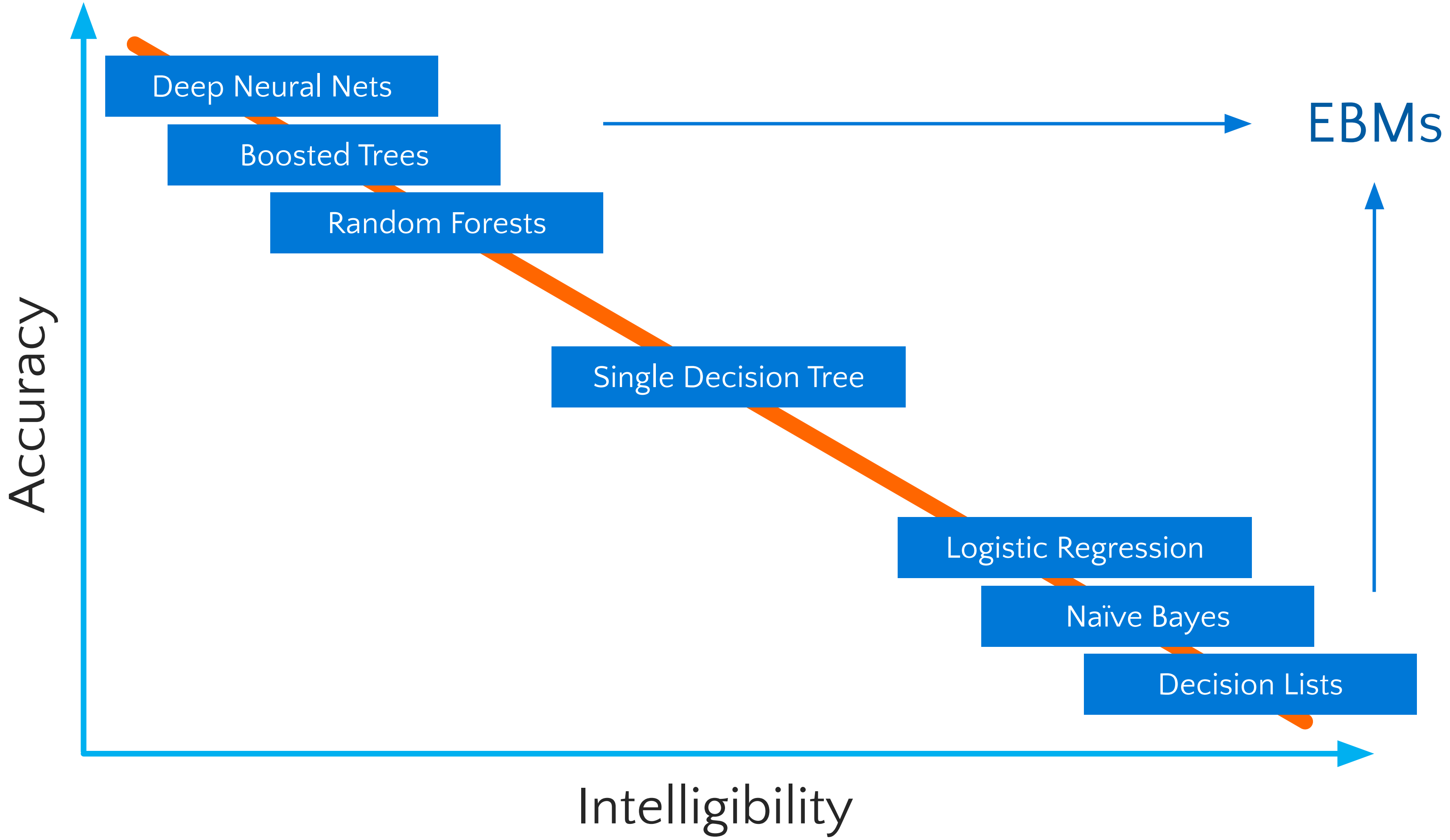
Accuracy vs. Intelligibility Tradeoff ???



Accuracy vs. Intelligibility Tradeoff -- Not True for Tabular Data

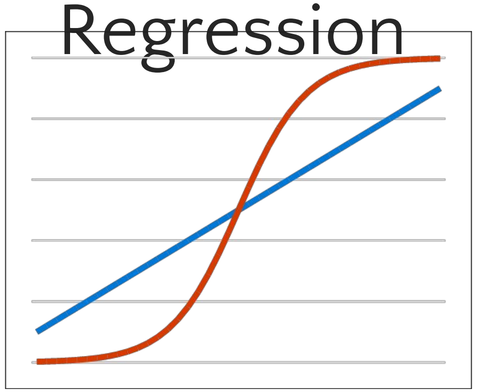


Accuracy vs. Intelligibility Tradeoff -- Not True for Tabular Data



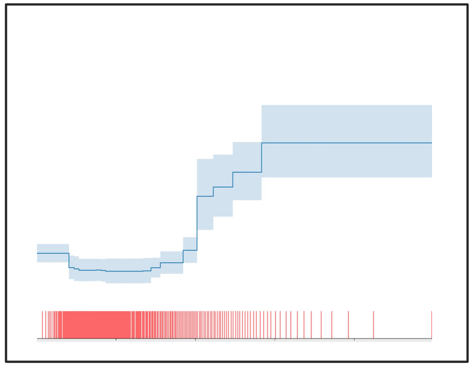
EBMs: Generalized Additive Models (GAMs)

Linear/Logisti
c
Regression



- Interpretable
- Not very accurate
- Can't model nonlinearities
- Can't model normal in middle
- Sometimes gets sign wrong

GAMs/EBMs



- More interpretable than linear/logistic
- Can be very accurate
- Can model nonlinearities
- Can model normal in middle
- More likely to show important effects

BlackBox
Machine
Learning

- Not interpretable (blackbox)
- Can be very accurate
- Can model nonlinearities
- Can model normal in middle
- Likely to learn spurious effects

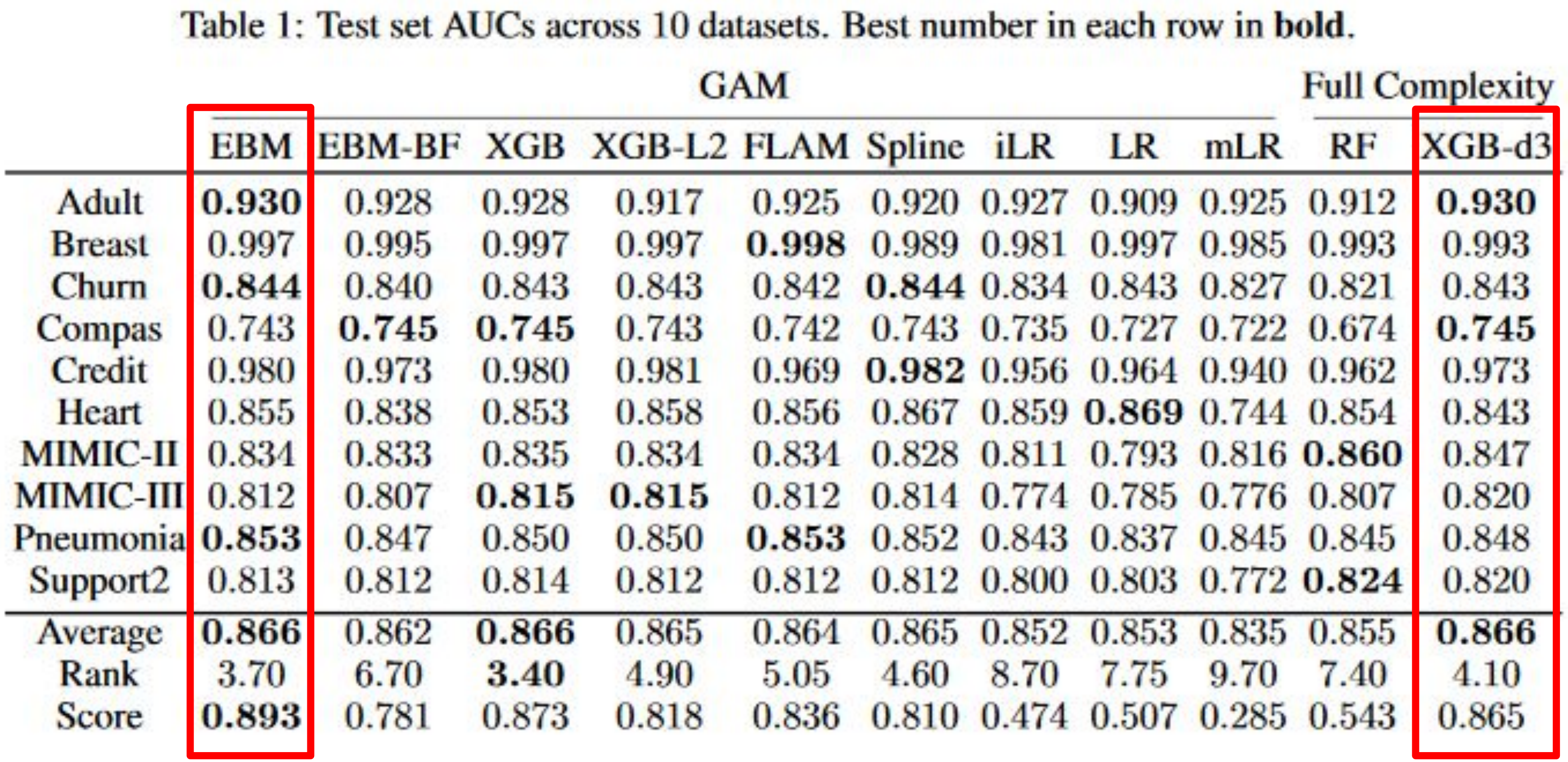
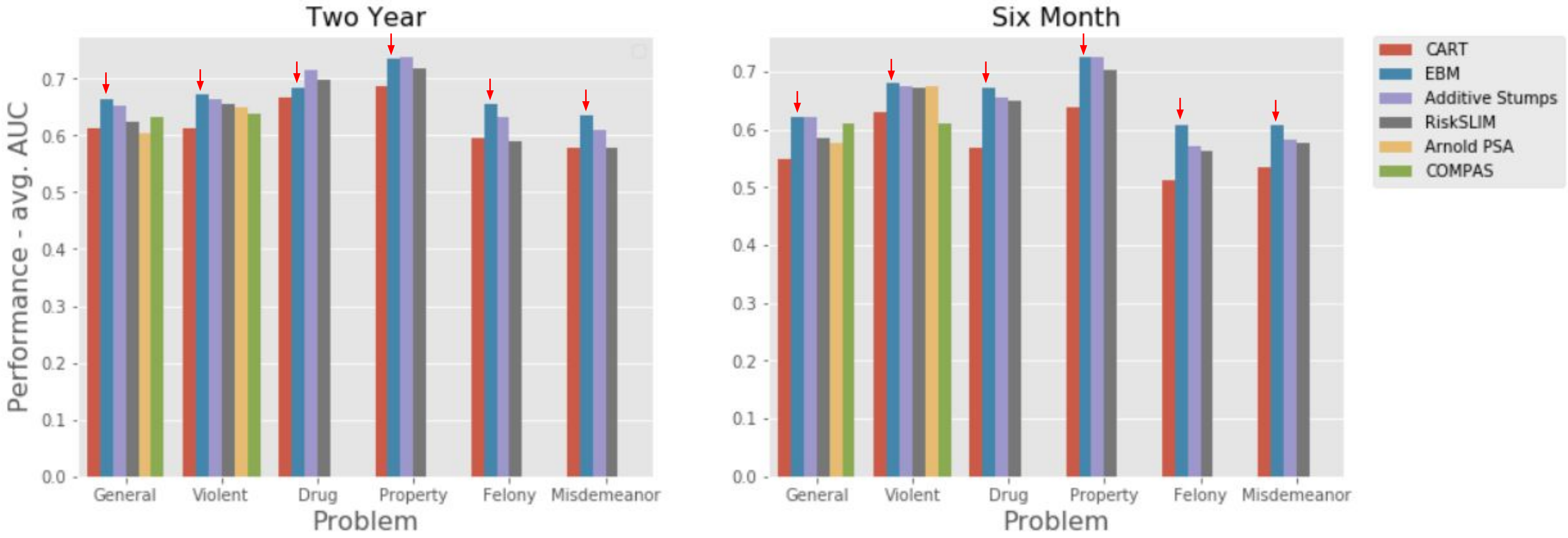


Table 1: Test set AUCs across 10 datasets. Best number in each row in **bold**.

	GAM									Full Complexity	
	EBM	EBM-BF	XGB	XGB-L2	FLAM	Spline	iLR	LR	mLR	RF	XGB-d3
Adult	**0.930**	0.928	0.928	0.917	0.925	0.920	0.927	0.909	0.925	0.912	**0.930**
Breast	0.997	0.995	0.997	0.997	**0.998**	0.989	0.981	0.997	0.985	0.993	0.993
Churn	**0.844**	0.840	0.843	0.843	0.842	**0.844**	0.834	0.843	0.827	0.821	0.843
Compas	0.743	**0.745**	**0.745**	0.743	0.742	0.743	0.735	0.727	0.722	0.674	**0.745**
Credit	0.980	0.973	0.980	0.981	0.969	**0.982**	0.956	0.964	0.940	0.962	0.973
Heart	0.855	0.838	0.853	0.858	0.856	0.867	0.859	**0.869**	0.744	0.854	0.843
MIMIC-II	0.834	0.833	0.835	0.834	0.834	0.828	0.811	0.793	0.816	**0.860**	0.847
MIMIC-III	0.812	0.807	**0.815**	**0.815**	0.812	0.814	0.774	0.785	0.776	0.807	0.820
Pneumonia	**0.853**	0.847	0.850	0.850	**0.853**	0.852	0.843	0.837	0.845	0.845	0.848
Support2	0.813	0.812	0.814	0.812	0.812	0.812	0.800	0.803	0.772	**0.824**	0.820
Average	**0.866**	0.862	**0.866**	0.865	0.864	0.865	0.852	0.853	0.835	0.855	**0.866**
Rank	3.70	6.70	**3.40**	4.90	5.05	4.60	8.70	7.75	9.70	7.40	4.10
Score	**0.893**	0.781	0.873	0.818	0.836	0.810	0.474	0.507	0.285	0.543	0.865

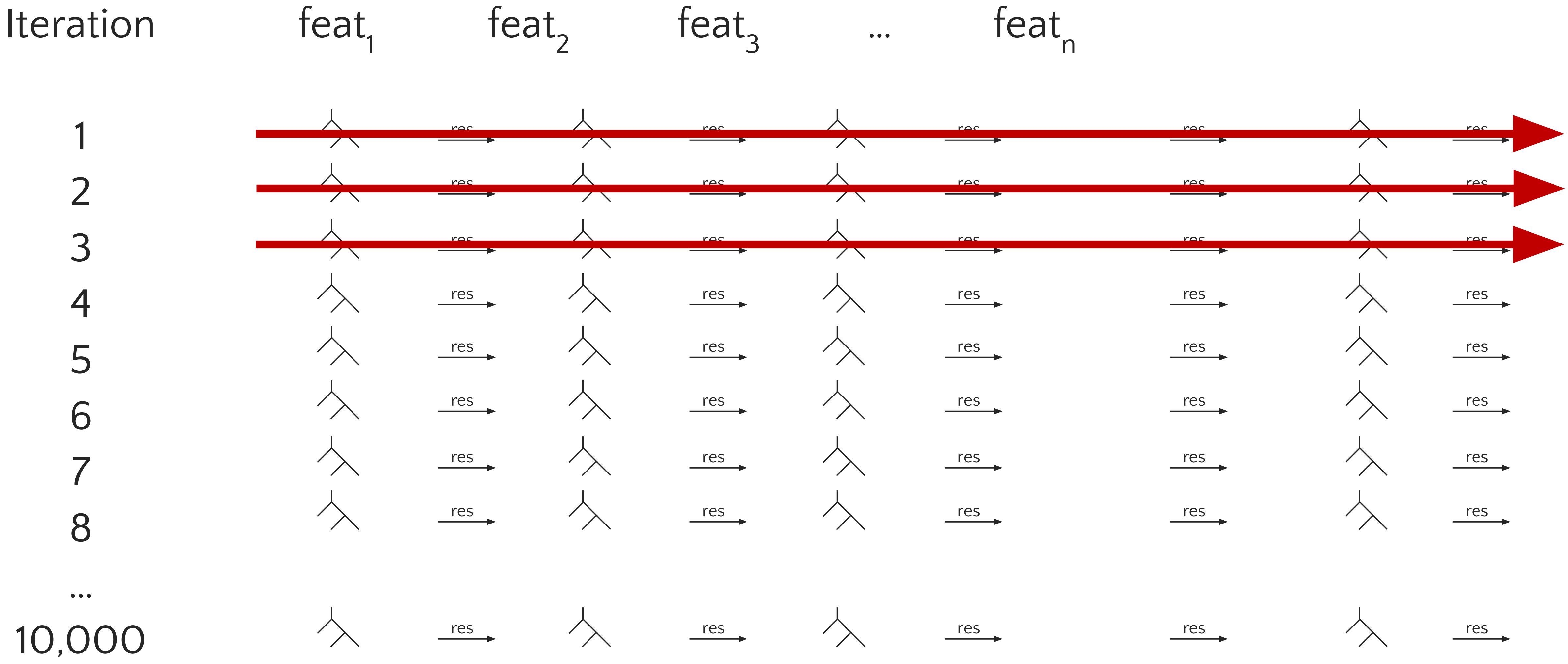
Chang, C.H., Tan, S., Lengerich, B., Goldenberg, A. and Caruana, R.
“How Interpretable and Trustworthy are GAMs?” *KDD2021*



“We observed that the best interpretable models can perform approximately as well as the best black-box models(XGBoost)”

Wang, C., Han, B., Patel, B., Mohideen, F. and Rudin, C., 2020.
In Pursuit of Interpretable, Fair and Accurate Machine Learning for Criminal Recidivism Prediction. *arXiv preprint arXiv:2005.04176*.

How Are EBMs Trained?



Iteration

$feat_1$ $feat_2$ $feat_3$ … $feat_n$

Iteration $\quad$ feat$_1$ $\quad$ feat$_2$ $\quad$ feat$_3$ $\quad$ … $\quad$ feat$_n$

1

2

3

4

5

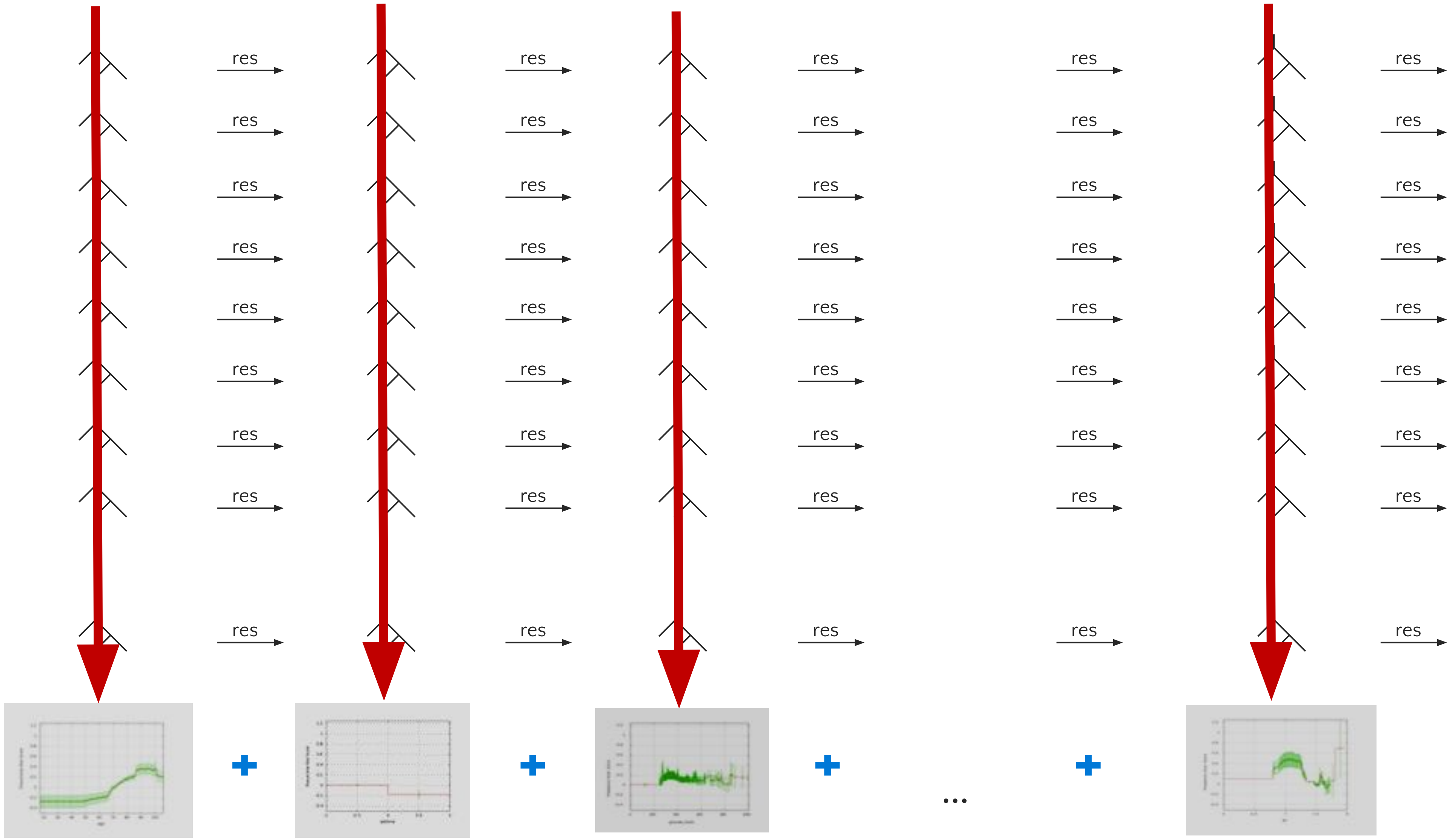
6

7

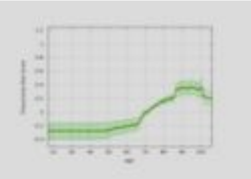
8

…

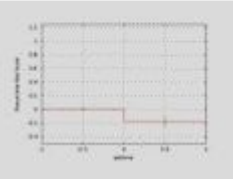
10,000



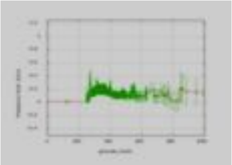
feat_1 feat_2 feat_3 … feat_n



+



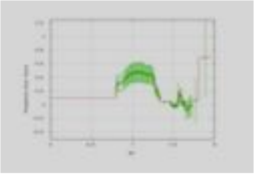
+



+

…

+



Limitations of EBMs

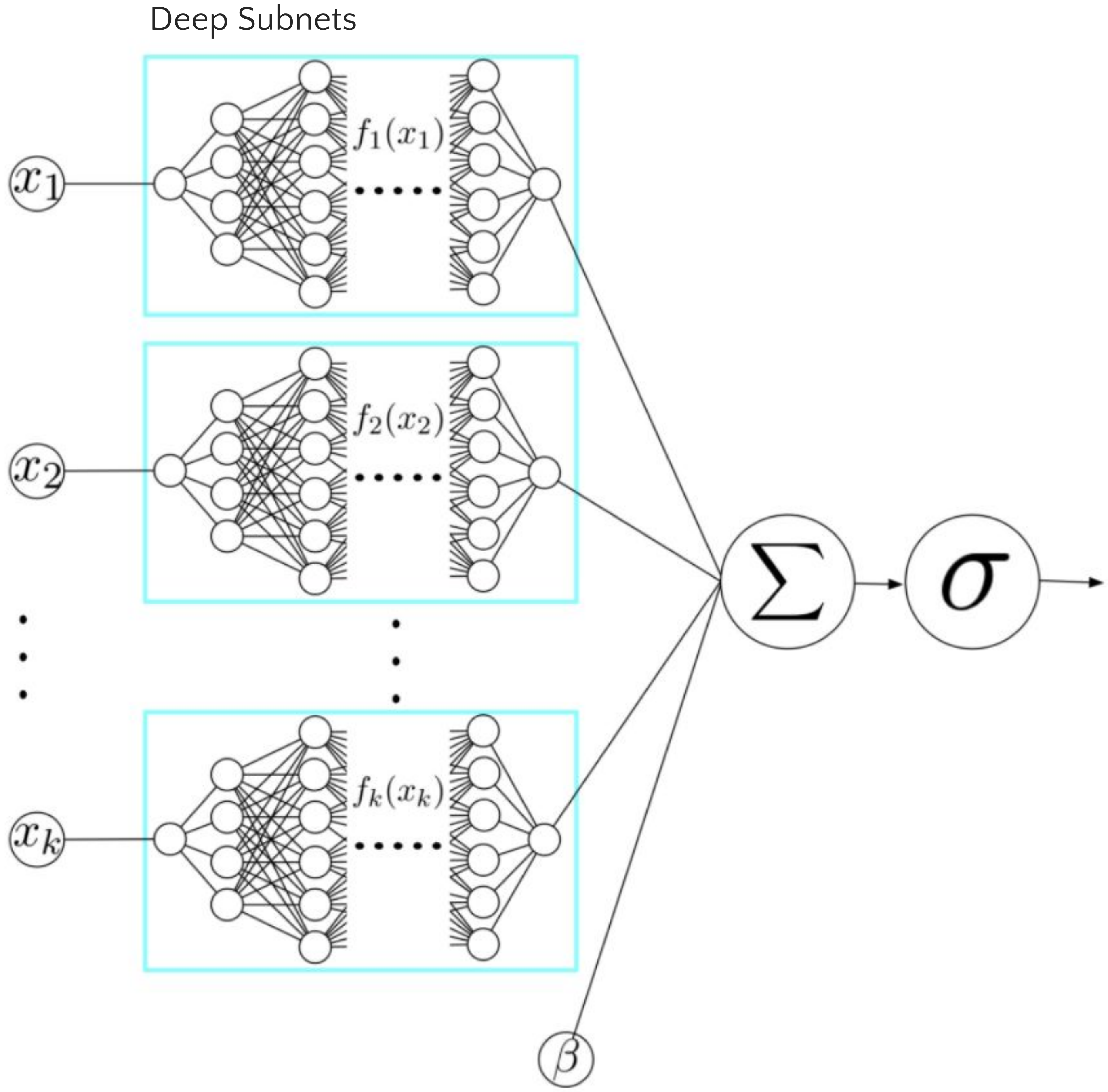
- EBMs have been state-of-the-art in glass-box learning for 5-10 years

- But…

- More than half of the ML community uses neural nets, not boosted trees
- Algorithms based on boosted trees don't scale as well as DNNs/CNNs trained on GPUs
- GAMs trained with boosted trees are not differentiable, which reduces flexibility
- Models trained with neural nets are much more modular and flexible
- Hard to make some things like multitask learning work with boosted trees

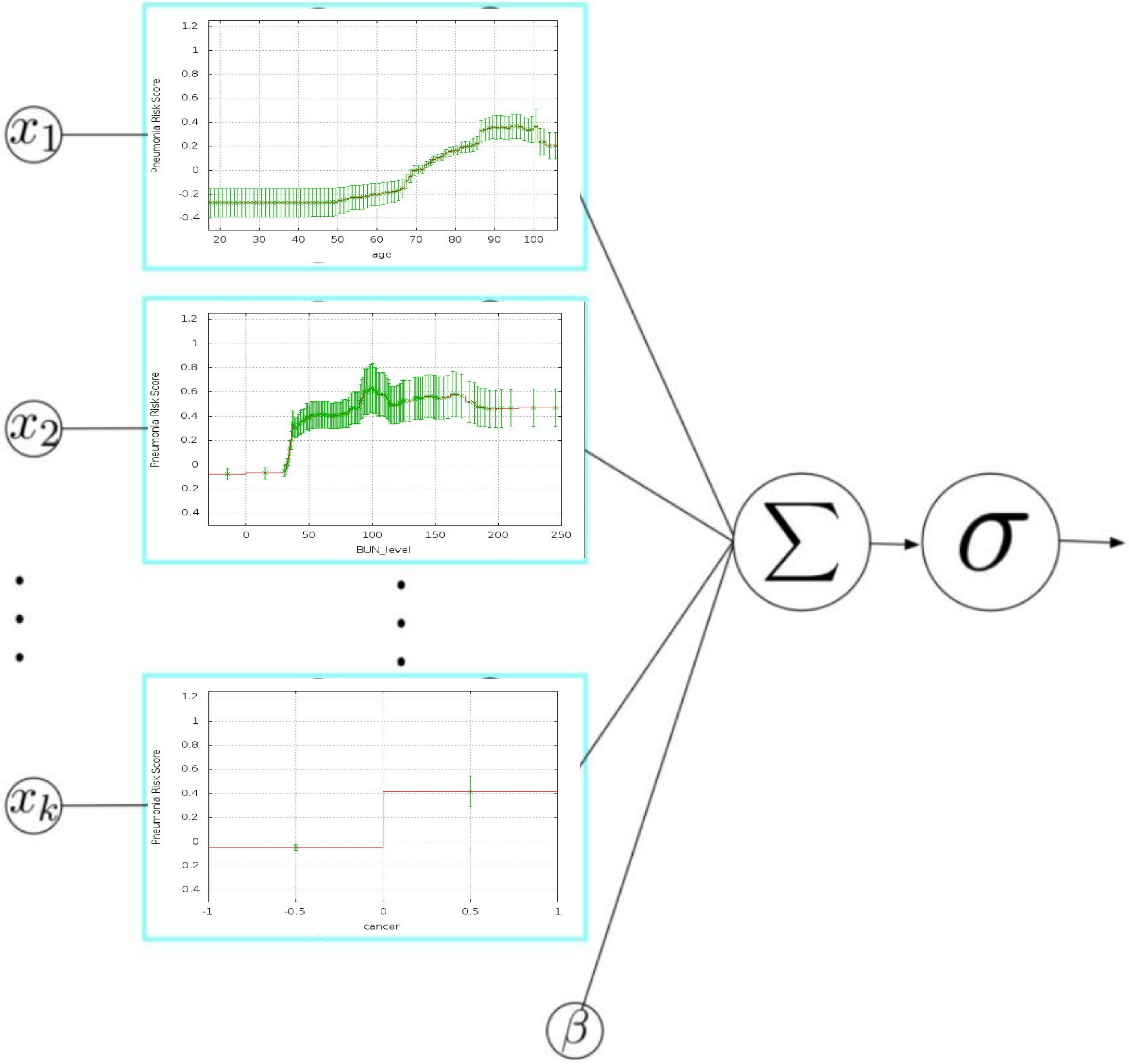
NAMs: Neural Additive Models

How Do We Fit GAMs with Neural Nets?



- Each feature feeds into a separate DNN subnet
- Subnets added at output layer
- Subnets learn separate additive models for each feature
- Sigmoid at output used for classification, not regression
- Subnets are learned in parallel
- Can be trained at massive scale on GPUs with standard software
- After training, subnets are replaced with graphs like EBMs

Deep Subnets □ Feature Graphs

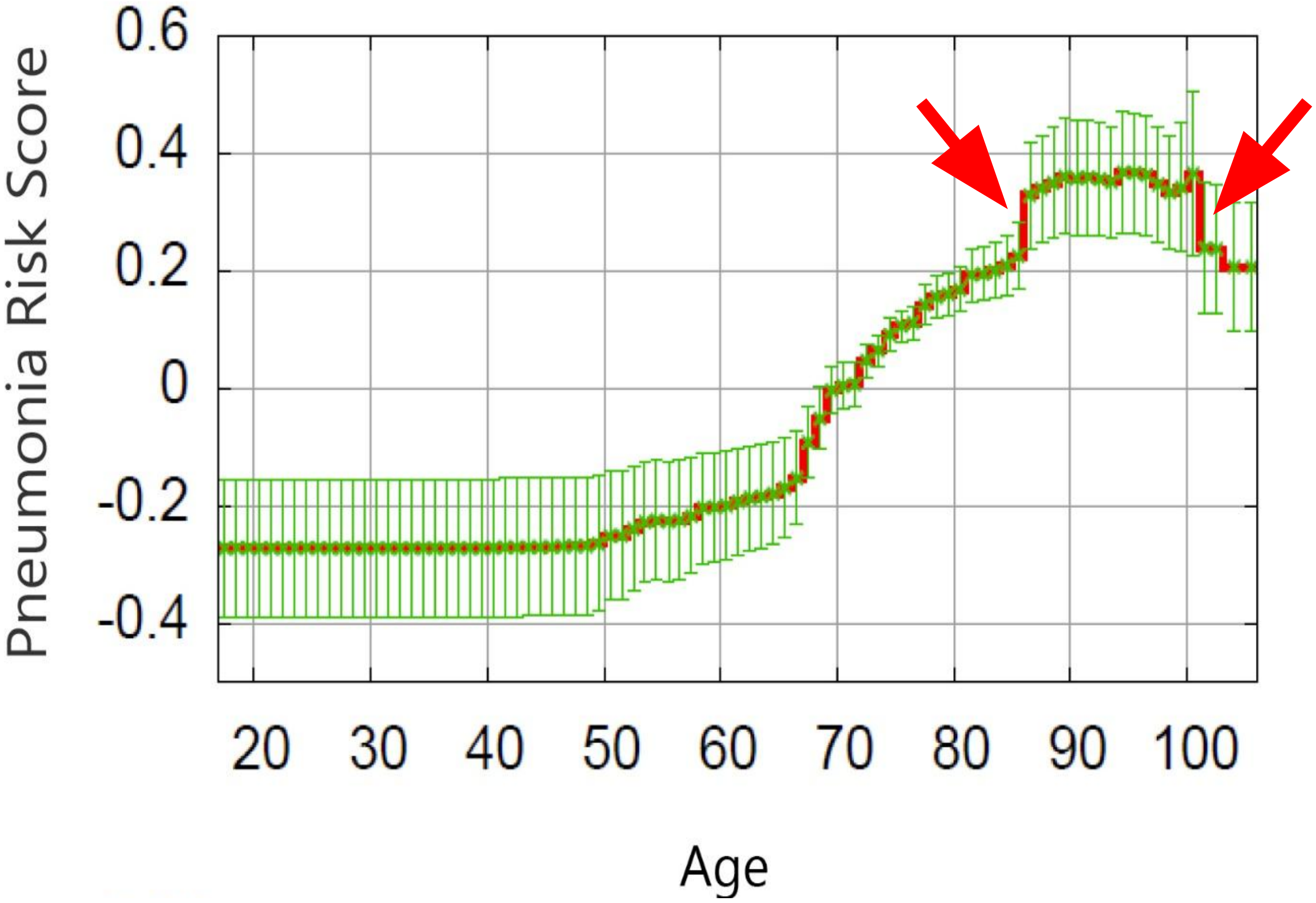


- Each feature feeds into a separate DNN subnet
- Subnets added at output layer
- Subnets learn separate additive models for each feature
- Sigmoid at output used for classification, not regression
- Subnets are learned in parallel
- Can be trained at massive scale on GPUs with standard software
- **After training, subnets are replaced with feature graphs**

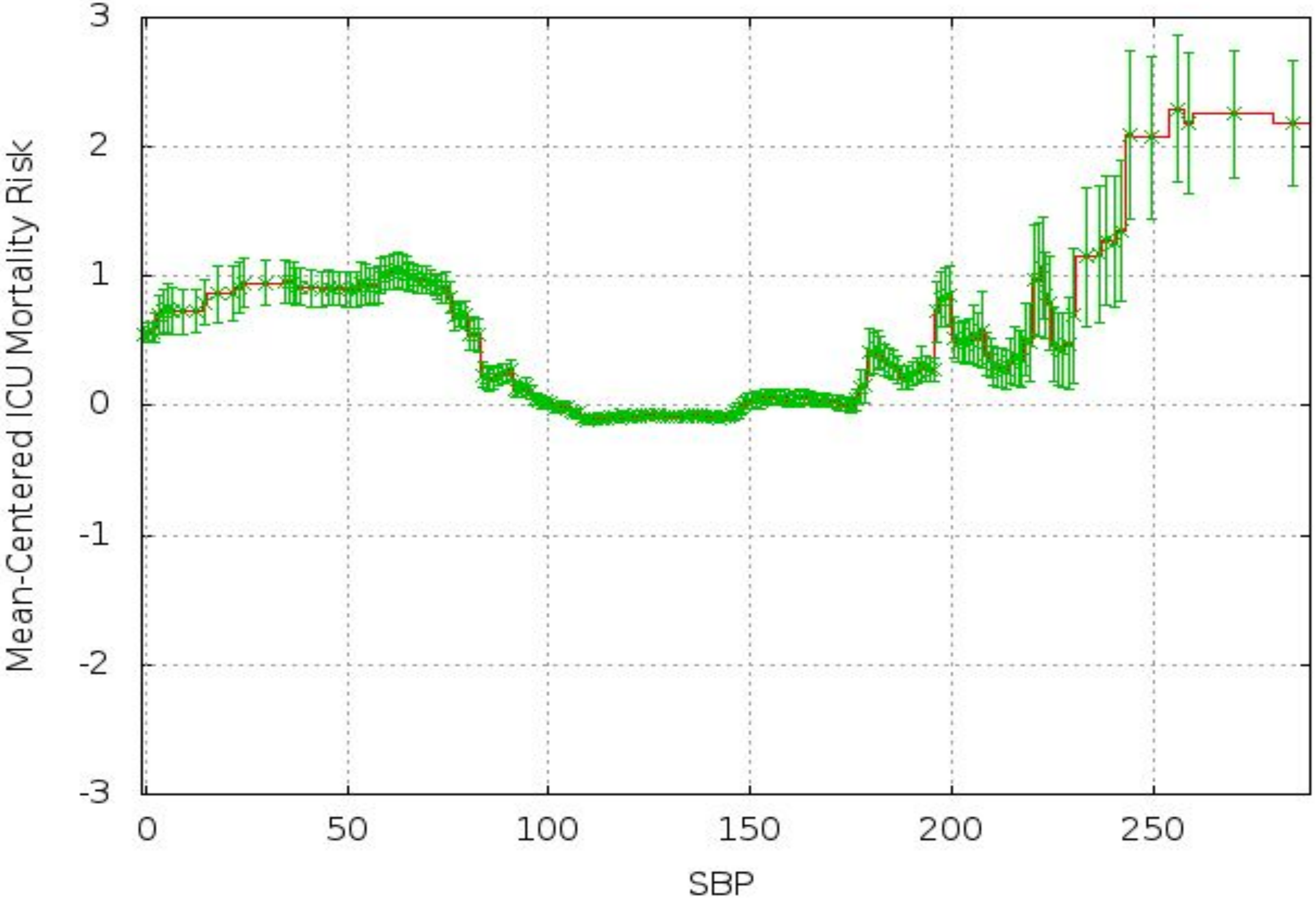
But there's a problem…

Work with EBMs Show Jumps in Graphs Are Important

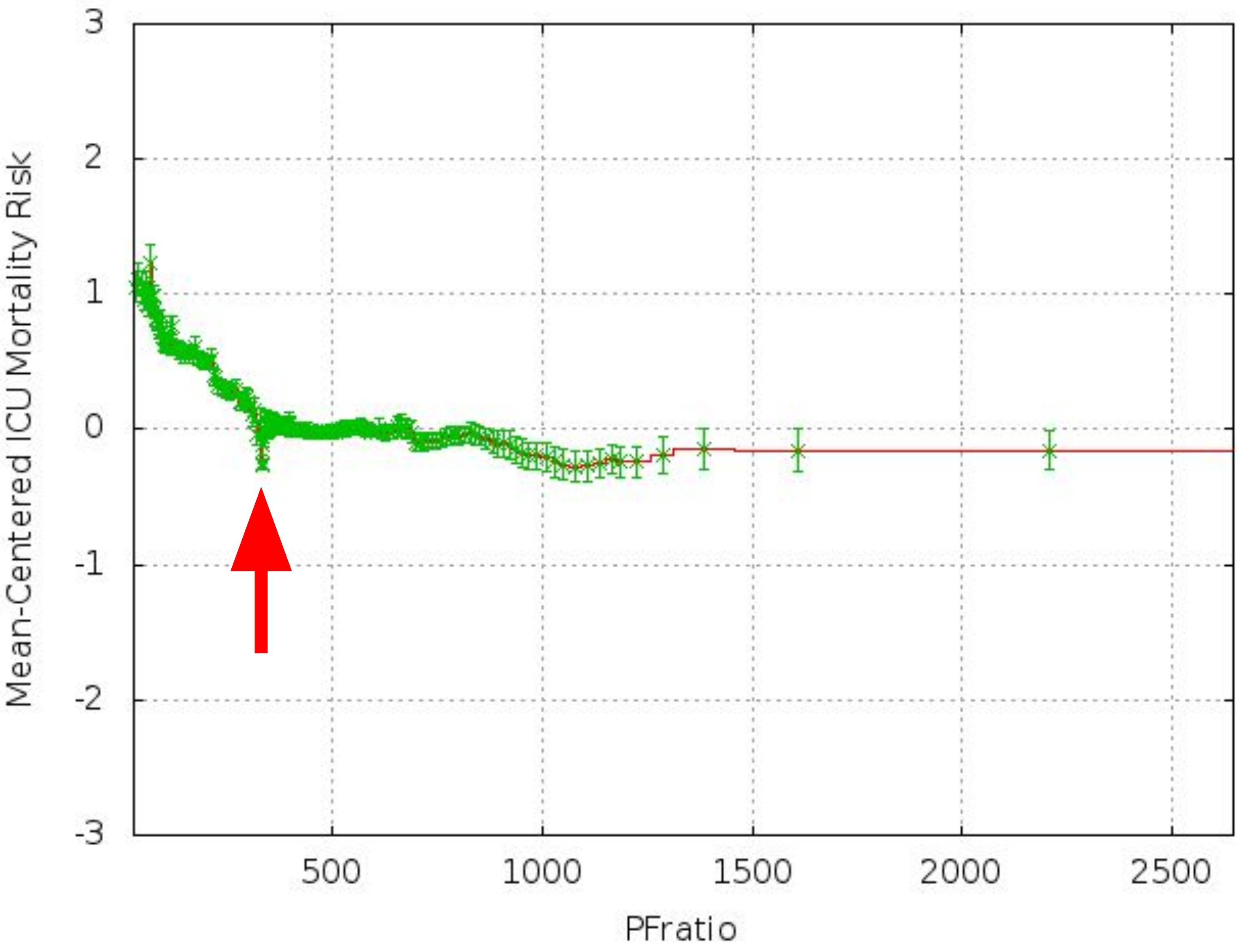
Work with EBMs Show Jumps in Graphs Are Important



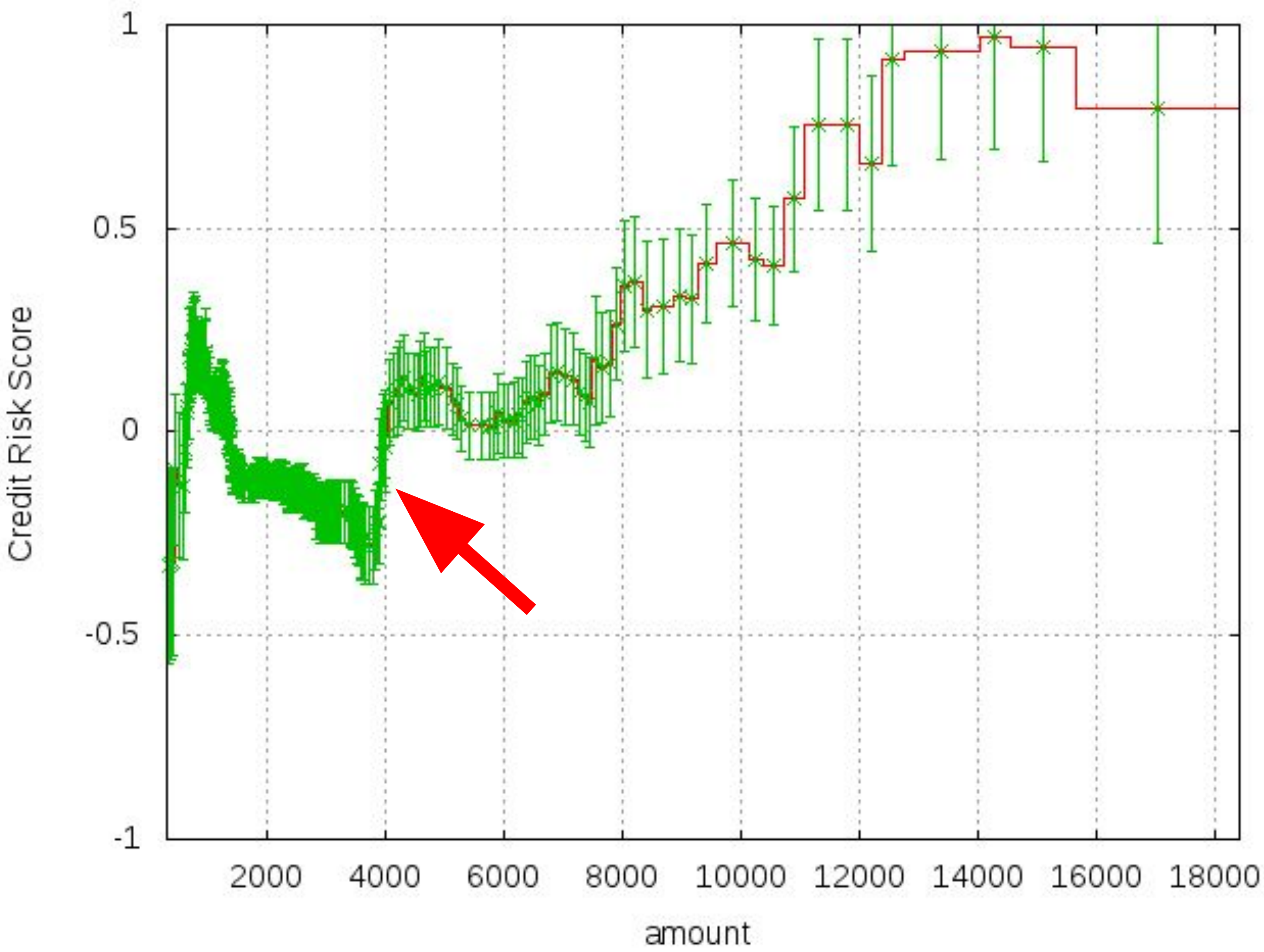
Work with EBMs Show Jumps in Graphs Are Important



Work with EBMs Show Jumps in Graphs Are

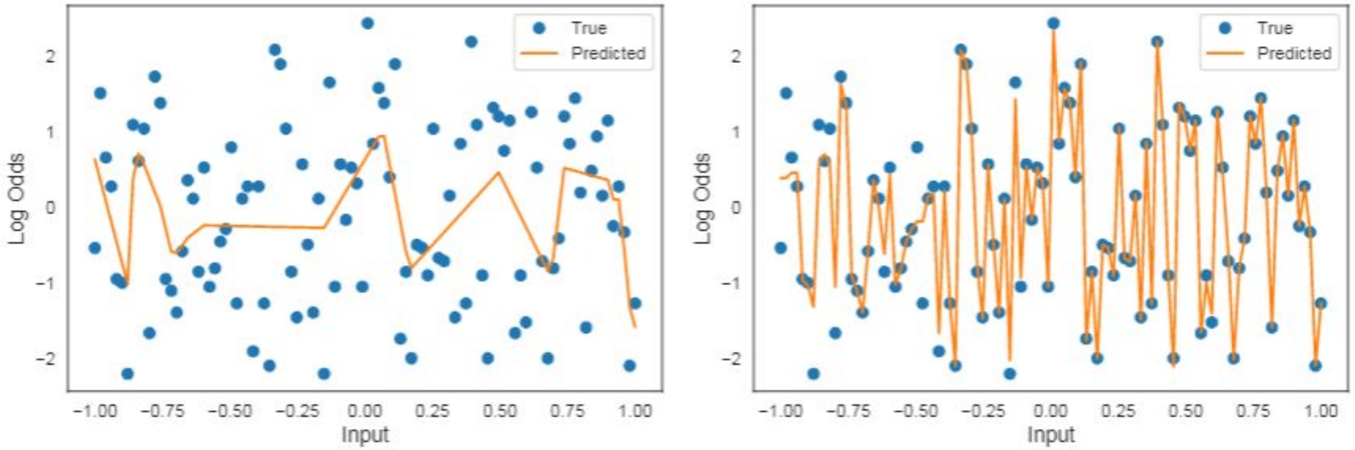


Work with EBMs Show Jumps in Graphs Are Important



DNNs Tend to Be Too Smooth to Learn Jumps Well

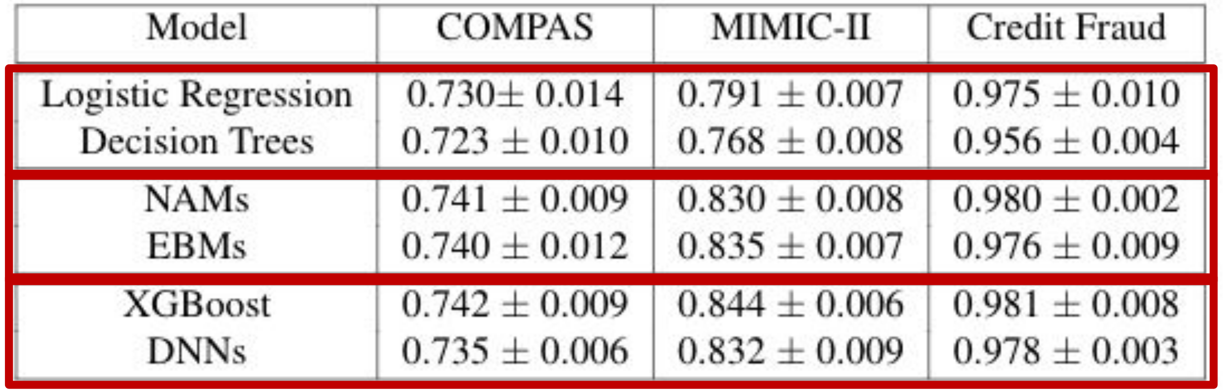
- How do we make DNNs “jumpier” without driving the entire model into overfitting?
- Trick is a special activation function: **ExU:** $h(x) = f\left(e^{w} * (x - b)\right)$
  - slope of activation function can be very steep so small changes in input => large changes in output



- Although overfitting is less of an issue in additive models like NAMs
  - To further reduce overfitting, we apply dropout, weight decay, capped ReLU activations, and also bag the NAM model 25–100 times to form an ensemble

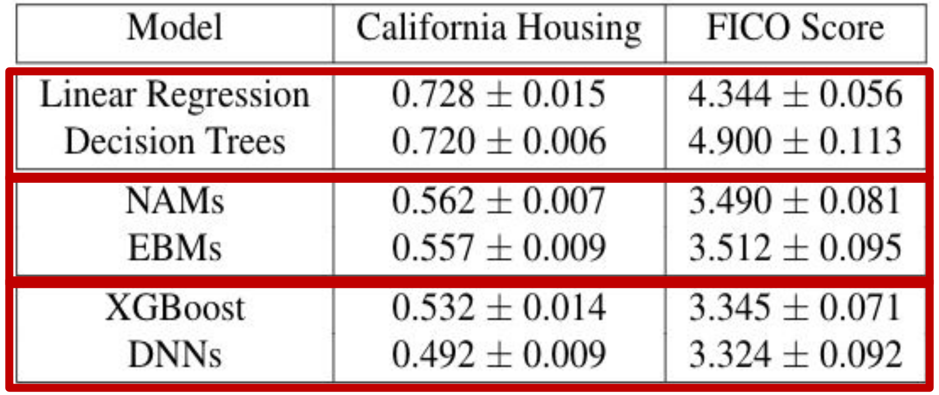
Empirical Results

Accuracy of NAMs



Model	COMPAS	MIMIC-II	Credit Fraud
Logistic Regression	0.730± 0.014	0.791 ± 0.007	0.975 ± 0.010
Decision Trees	0.723 ± 0.010	0.768 ± 0.008	0.956 ± 0.004
NAMs	0.741 ± 0.009	0.830 ± 0.008	0.980 ± 0.002
EBMs	0.740 ± 0.012	0.835 ± 0.007	0.976 ± 0.009
XGBoost	0.742 ± 0.009	0.844 ± 0.006	0.981 ± 0.008
DNNs	0.735 ± 0.006	0.832 ± 0.009	0.978 ± 0.003

AUC on classification datasets.
Higher is better.

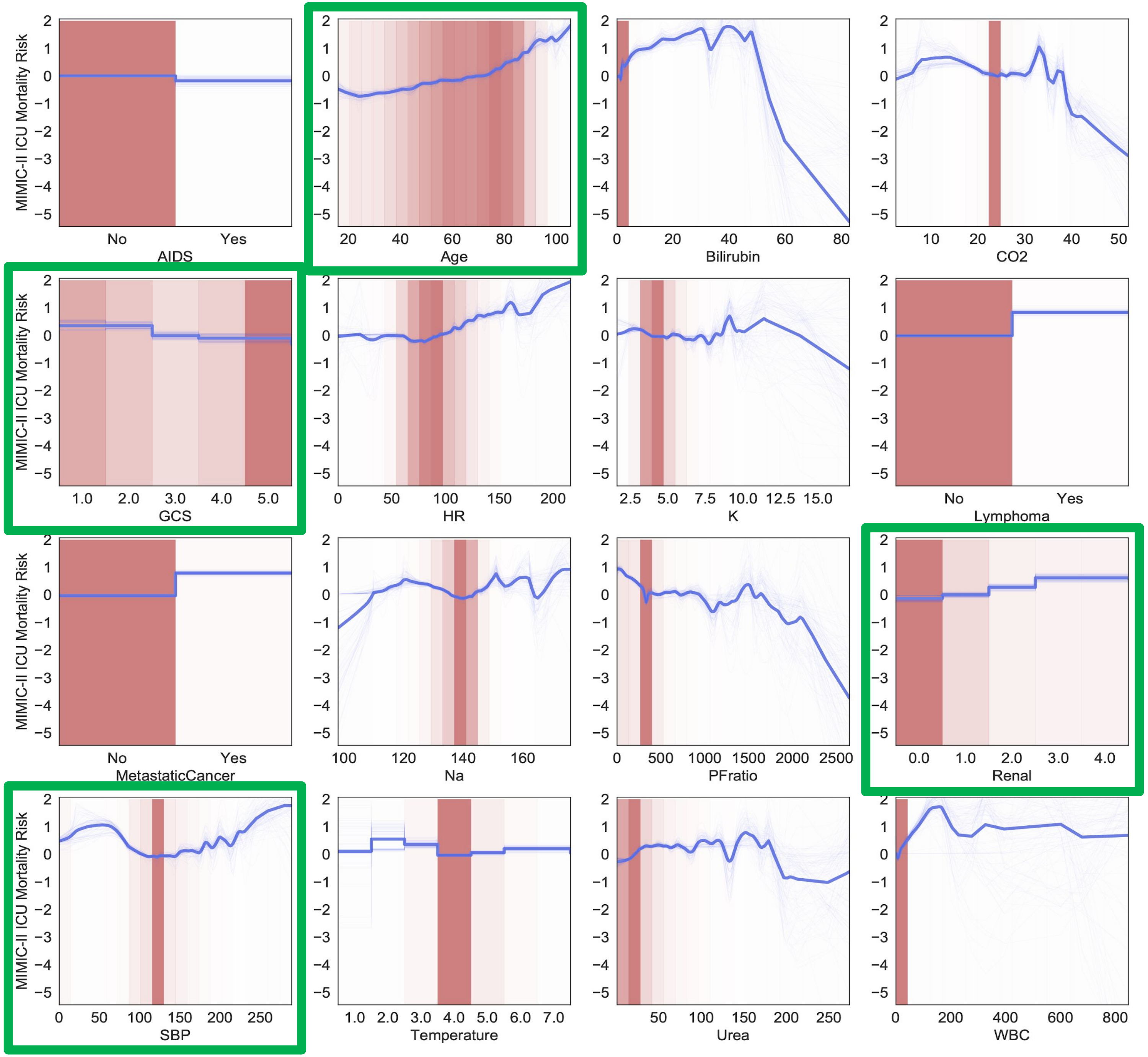


Model	California Housing	FICO Score
Linear Regression	0.728 ± 0.015	4.344 ± 0.056
Decision Trees	0.720 ± 0.006	4.900 ± 0.113
NAMs	0.562 ± 0.007	3.490 ± 0.081
EBMs	0.557 ± 0.009	3.512 ± 0.095
XGBoost	0.532 ± 0.014	3.345 ± 0.071
DNNs	0.492 ± 0.009	3.324 ± 0.092

RMSE on regression datasets.
Lower is better.

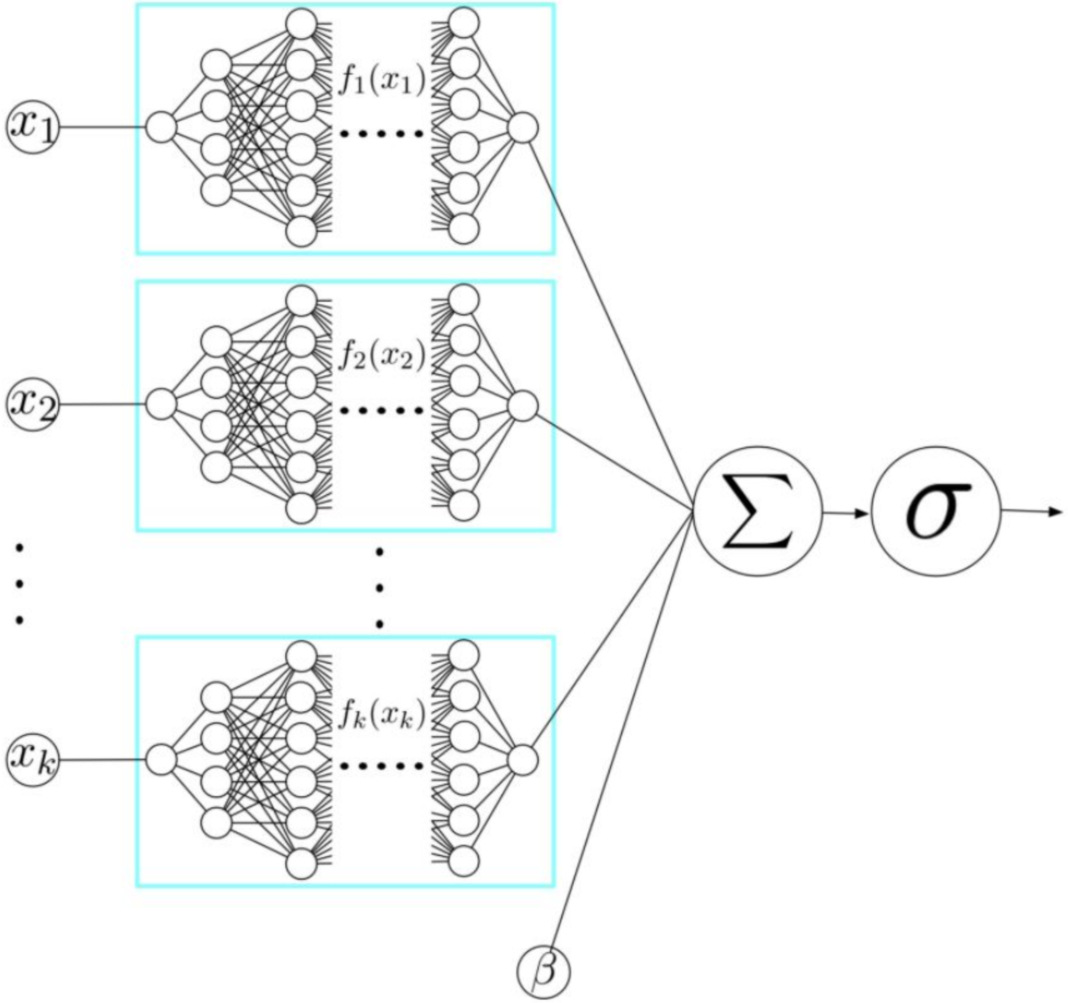
A little loss in accuracy for NAMs
compared to DNNs on tabular data!

MIMIC-II ICU Mortality Prediction

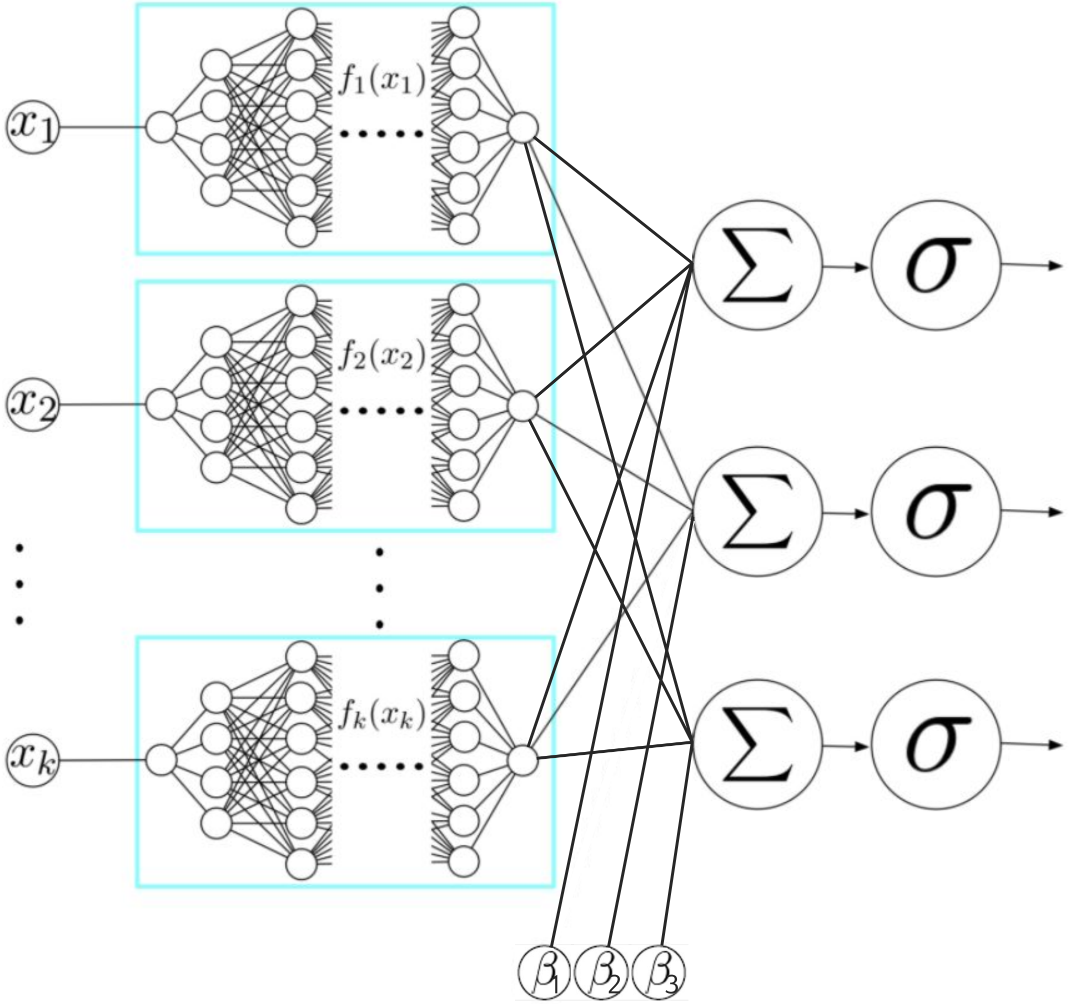


Multitask Learning with NAMs

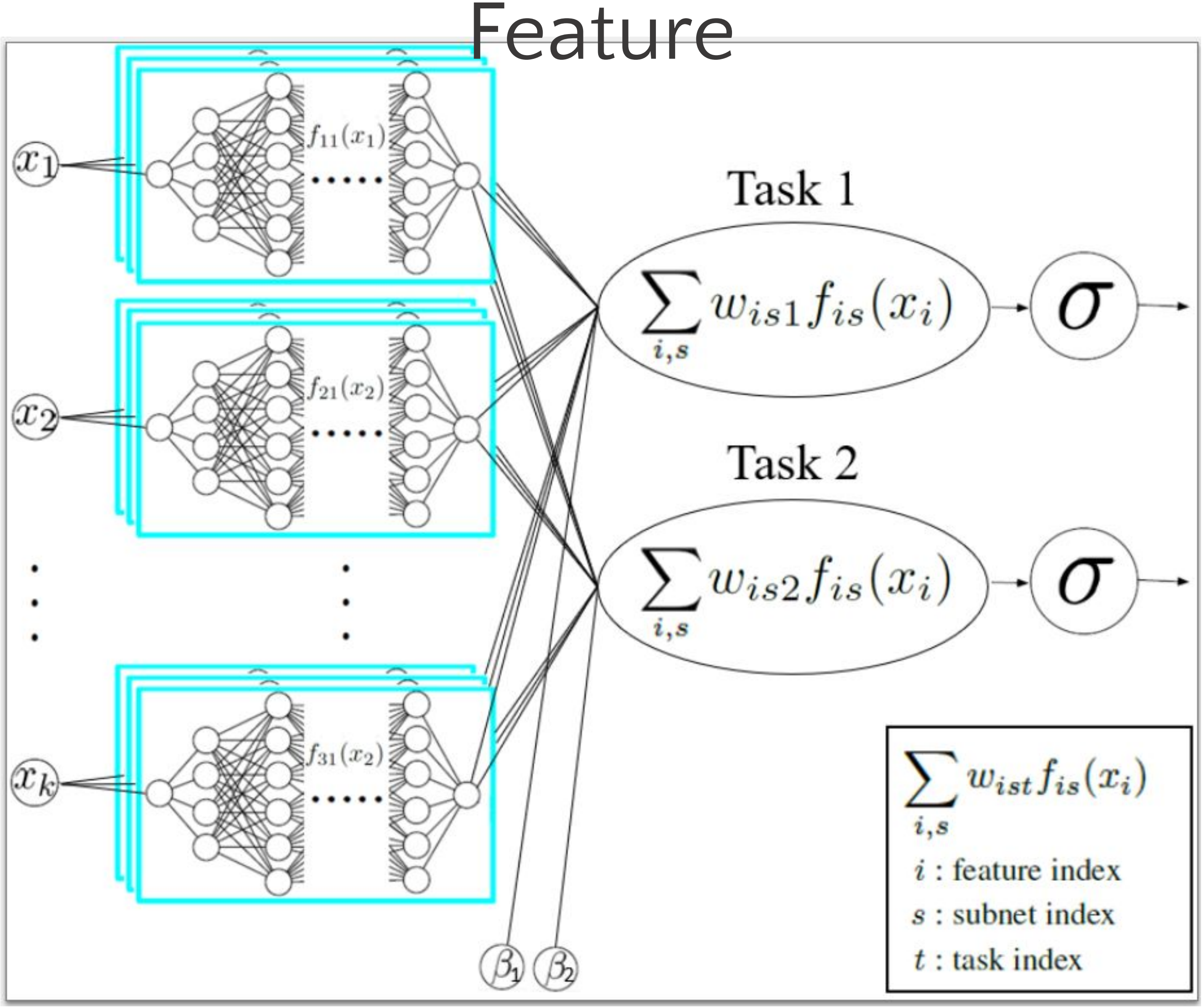
Single Task NAM

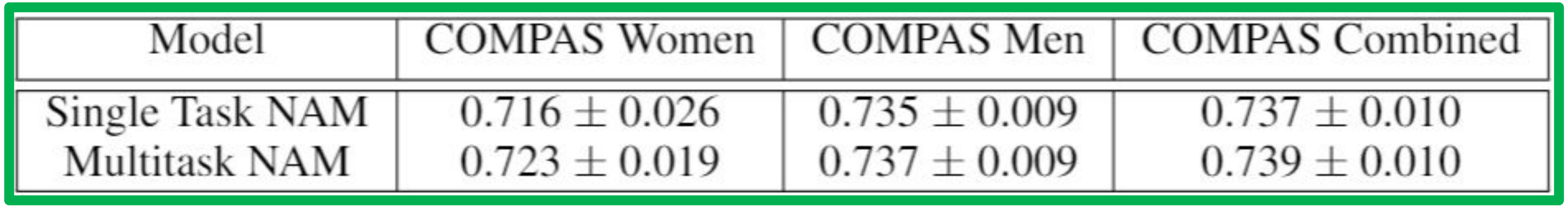


MultiTask NAM

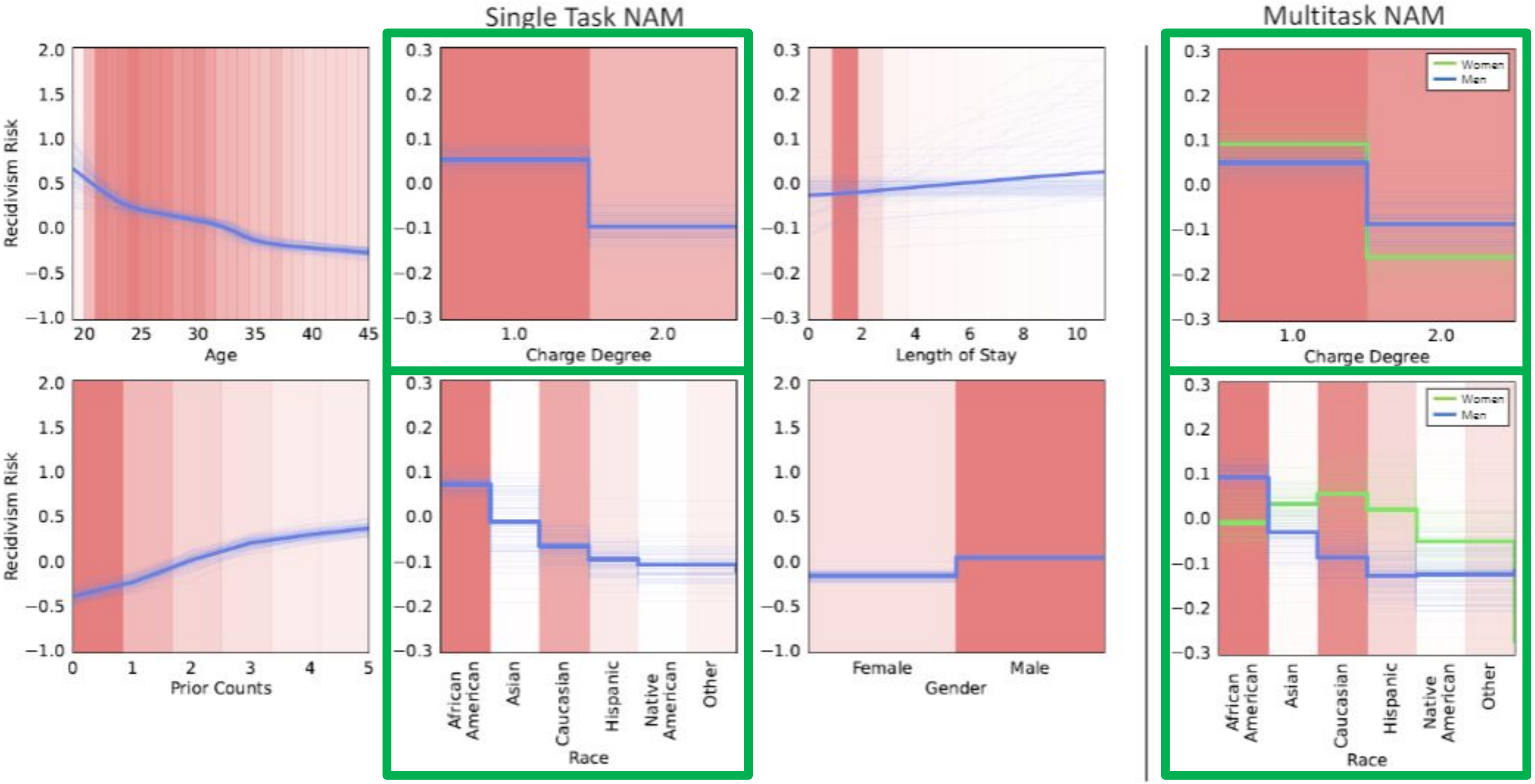


More Flexible MultiTask NAM: Multiple SubNets per Feature



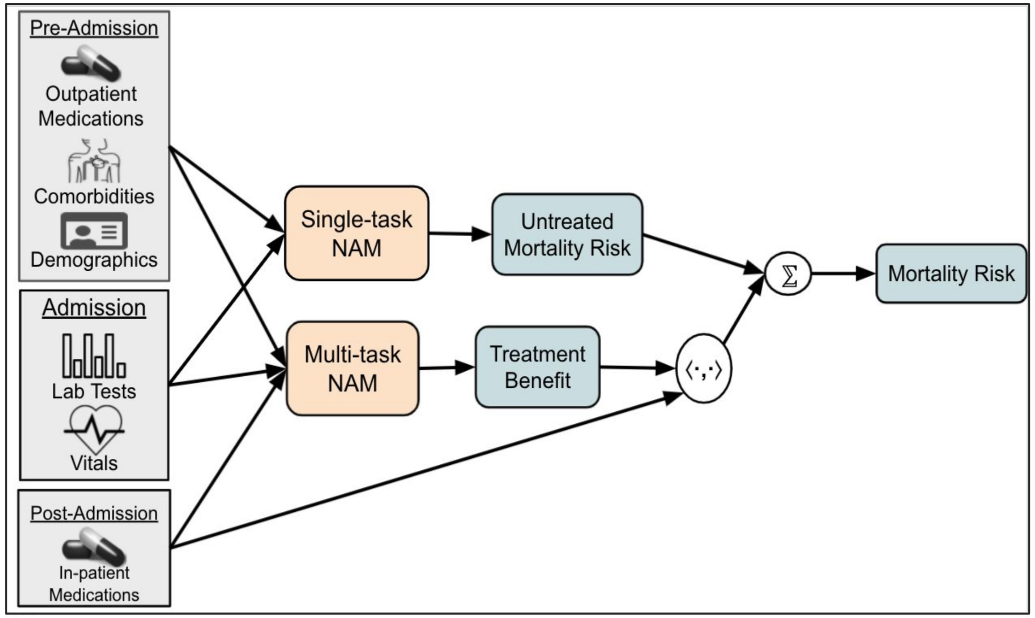


Model	COMPAS Women	COMPAS Men	COMPAS Combined
Single Task NAM	0.716 ± 0.026	0.735 ± 0.009	0.737 ± 0.010
Multitask NAM	0.723 ± 0.019	0.737 ± 0.009	0.739 ± 0.010



Benefitting from Differentiability & MultiTask Learning

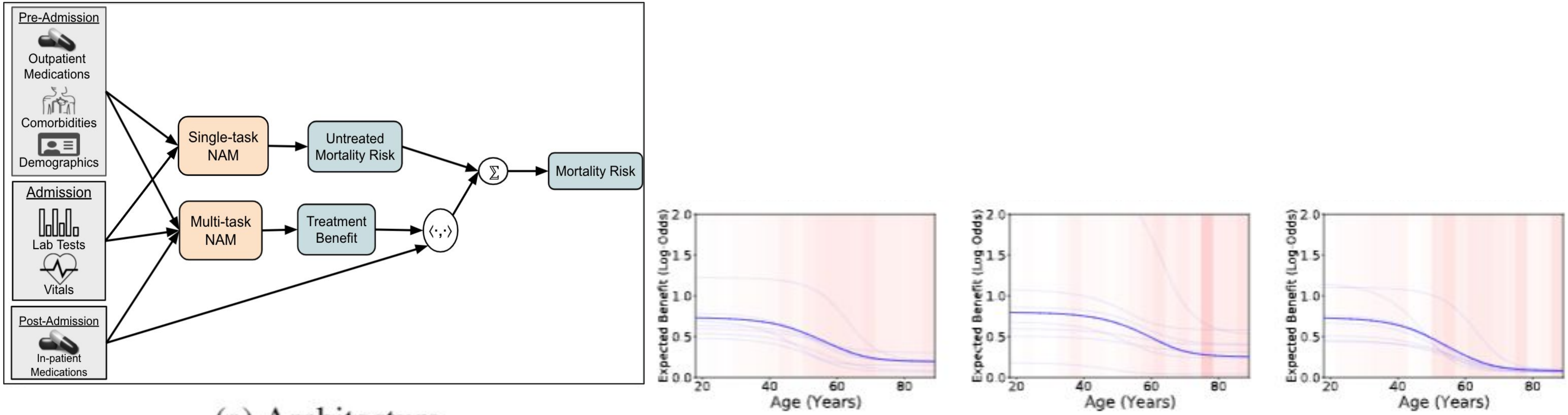
Estimating Personalized Treatment Benefits for COVID-19



(a) Architecture

Lengerich et al. Automated Interpretable Discovery of Variable Treatment Effectiveness: A Covid-19 Case Study. 2021.

Estimating Personalized Treatment Benefits for COVID-19



(a) Architecture

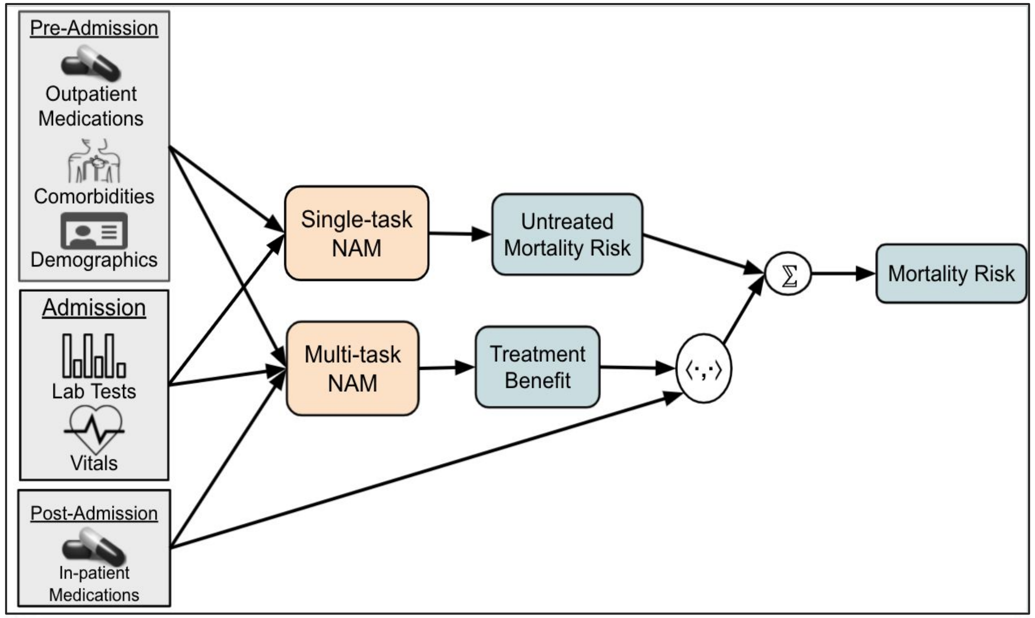
(b) Anti-Coagulants

(c) NSAIDs

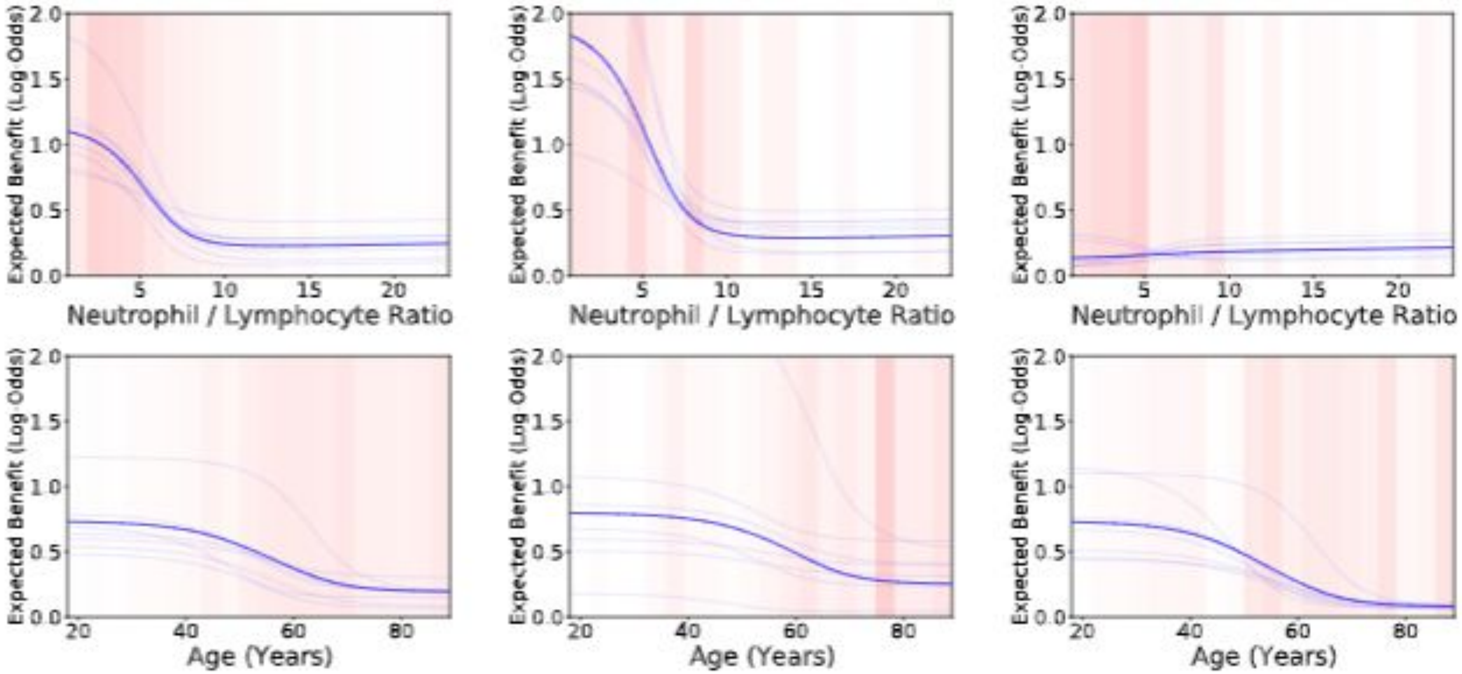
(d) Glucocorticoids

Lengerich et al. Automated Interpretable Discovery of Variable Treatment Effectiveness: A Covid-19 Case Study. 2021.

Estimating Personalized Treatment Benefits for COVID-19



(a) Architecture



(b) Anti-Coagulants

(c) NSAIDs

(d) Glucocorticoids

Lengerich et al. Automated Interpretable Discovery of Variable Treatment Effectiveness: A Covid-19 Case Study. 2021.

Summary

- Glassbox learning can be as accurate as Blackbox learning on Tabular Data
  - Accurate
  - Interpretable
  - Editable
- NAMs allow us to train state-of-the-art GAMs with Deep Neural Nets
  - Fully interpretable and editable
  - Differentiable
  - More flexible & modular: multitask learning, more complex architectures like personalized medicine
  - Can scale because they can be trained GPUs
- Building easy-to-use toolkits so everyone can train GAMs
- Many opportunities going forward to combine NAMs with DNNs, CNNs, RI, …