Can We Learn to Explain Chest X-Rays?: A Cardiomegaly Use Case

Neil Jethani[1,2], **Mukund Sudarshan**[2], **Lea Azour**[3], **William Moore**[3], **Yindalon Aphinyanaphongs**[1], **Rajesh Ranganath**[1,2,4]
[1]Department of Population Health, NYU Grossman School of Medicine
[2]Courant Institute of Mathematical Sciences, New York University
[3]Department of Radiology, NYU Grossman School of Medicine
[4]Center for Data Science, New York University
{neil.jethani, yin.a, william.moore, lea.azour}@nyulangone.org
{sudarshan, rajeshr}@cims.nyu.edu

1 Introduction

Machine learning (ML) is routinely used to analyze medical images in radiology [4, 9, 24], ophthalmology [6, 11], cardiology [10, 2], and pathology [5, 7]. To effectively capitalize on these advances, clinicians need to be able to interpret why clinical decisions are made by ML models. These interpretations can help clinicians achieve three critical milestones — the ability to trust model decisions [15], identify model failure modes [26], and expand clinical knowledge[20].

There are four general approaches for providing interpretability — perturbation-based, gradient-based, locally linear, and amortized explanation methods (AEMs). Perturbation-based approaches, such as [27, 28, 29], rely on computationally expensive perturbations of the input. Gradient-based methods [21, 22], such as grad-CAM [19], which aim to calculate the gradient of the target with respect to features in the input, have been shown to lack fidelity [12, 1]. Meanwhile, locally linear methods, popularly LIME [18] and SHAP [16], require learning a new model for each sample of data and rely on biased linear explanations. AEMs, L2X [3] and INVASE [25], are the only class of methods that provide an objective to measure explanation fidelity and global model to quickly explain any sample of data with a single forward pass. Given the need for high fidelity, real-time explanations in clinical care settings, we consider AEMs as the preferred approach for interpreting medical images and focus on addressing issues with prior AEMs.

Figure 1: **L2X classifies digits with** 96% **accuracy from a single selected pixel.**

Existing AEMs, L2X [3] and INVASE [25], learn a global selector model to select important feature subsets in concert with a predictor model to predict the target given this subset of features. While clinicians can benefit from using AEMs, we show that they should not trust existing these existing methods. Figure 1 illustrates explanations from L2X on MNIST [14] trained with a selector model that outputs a *single* important pixel and achieves 96.0% accuracy. Here, the selector model makes the classification decision and transmits it to the predictor model through the binary code of the selector variables.

We show that this phenomenon can occur even with simple medical imaging tasks, such as predicting cardiomegaly. We propose REAL-X, a new AEM, which addresses the issues with prior AEMs by respecting the true data generating distribution. We show that our method provides trustworthy explanations through quantitative and *expert radiologist* evaluation.

2 Instance-wise feature selection (IWFS)

Let input $\mathbf{x}$ be a random vector in $\mathbb{R}^D$, and the target $\mathbf{y} \in \{1, \ldots, K\}$. We refer to the jth component of $\mathbf{x}$ as $\mathbf{x}_j$, and a subset of features as $\mathbf{x}_\mathcal{S} := \{\mathbf{x}_j\}_{j \in \mathcal{S}}$, where $\mathcal{S} \subseteq \{1, \ldots, D\}$. F is a distribution over $(\mathbf{x}, \mathbf{y})$. *Instance-wise feature selection (*IWFS*)*, for every instance $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \sim F(\mathbf{x}, \mathbf{y})$, seeks to identify a minimal subset of features $\boldsymbol{x}^{(i)}_{\mathcal{S}^{(i)}}$ such that the following condition is met:

$$F(\mathbf{y} \mid \mathbf{x}_{\mathcal{S}^{(i)}} = \boldsymbol{x}^{(i)}_{\mathcal{S}^{(i)}}) = F(\mathbf{y} \mid \mathbf{x} = \boldsymbol{x}^{(i)}). \tag{1}$$

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

2.1 Amortized explanation method (AEM)

AEMs refer to a general class of interpretability methods that learn a *global* selector model to identify a subset of important features *locally* in any given instance of data. The selector model is a distribution $q_{\text{sel}}(\mathbf{s} \mid \mathbf{x}; \beta)$ over a selector variable $\mathbf{s}$, which indicates the important features for a given sample of $\mathbf{x}$. AEMs optimize q_{sel} with an objective that measures the ability of the selections to predict the target. Existing AEMs, L2X [3] and INVASE [25], learn $q_{\text{sel}}(\mathbf{s} \mid \mathbf{x}; \beta)$ in concert with a predictor model $q_{\text{pred}}(\mathbf{y} \mid m(\mathbf{x}, \mathbf{s}); \theta)$. We refer to such methods as *joint amortized explanation methods* (JAMs). JAMs use a regularizer $R(\mathbf{s})$ to control the number of selected features and a masking function m to hide the jth feature $\boldsymbol{x}_j$ with the selector variable $\boldsymbol{s}_j$. To learn the parameters of the amortized selector model, β, and the predictor model, θ, the JAM objective maximizes

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\mathbb{E}_{\boldsymbol{s}\sim q_{\text{sel}}(\mathbf{s} \mid \boldsymbol{x};\beta)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}); \theta) - \lambda R(\boldsymbol{s})\right]. \tag{2}$$

We show that both L2X and INVASE fit this objective in appendix A.

3 JAMs Simply Encode Predictions

While the selector model in JAMs makes it simple to explain new examples, in this section, we reveal that JAMs can easily encode predictions. For noise free classification, the following lemma states that the selector model can encode the target using the selection of at most a single feature in each sample of data. For simplicity, we focus on independent Bernoulli selector variables $\mathbf{s}_j \sim \text{Bernoulli}(f_\beta(\mathbf{x})_j)$. The intuition here is that the selector variable $\mathbf{s}$ is a binary code that can pass quite a bit of information to predict the target. A proof is available in appendix E.1.

Lemma 1. *Let $\mathbf{x} \in \mathbb{R}^D$ and target $\mathbf{y} \in \{1, ..., K\}$. If $\mathbf{y}$ is a deterministic function of $\mathbf{x}$ and $K \leq D$, then* JAMs *with monotone increase regularizers R will select at most one feature at optimality.*

This idea can be generalized to settings where $\mathbf{y}$ is not a deterministic function of $\mathbf{x}$ and is formally captured in lemma 2 found in appendix D and proved in in appendix E.2.

4 REAL-X, Let Us Evaluate and Explain!

Evaluate. In order to trust the explanations provided by AEMs, selections need to be quantitatively evaluated. The goal of IWFS is to satisfy the condition in eq. (1). The evaluation of IWFS should reflect the goal—the selections should be evaluated on the true conditional distribution $F(\mathbf{y} \mid \mathbf{x}_{S^{(i)}} = \boldsymbol{x}^{(i)}_{S^{(i)}})$. More generally, evaluating the selection of any potential subset of features $\mathcal{R}$ requires access to $F(\mathbf{y} \mid \mathbf{x}_\mathcal{R})$. This distribution can be estimated with q_{eval}, trained by maximizing the follow objective, which yields the true $F(\mathbf{y} \mid \mathbf{x}_\mathcal{R})$ at optimality (see appendix F):

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\mathbb{E}_{\boldsymbol{r}\sim\text{Bernoulli}(0.5)}\left[\log q_{\text{eval}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{r}); \eta)\right]. \tag{3}$$

Explain. Now we describe a method to ensure that the learned selections also respect the true data distribution given subsets of the input $F(\mathbf{y} \mid \mathbf{x}_\mathcal{R})$. JAMs learn to select features and make predictions in concert. This flexibility allow JAMs to learn to make predictions from information encoded in the choice of selections. We propose learning the predictor model disjointly to approximate $F(\mathbf{y} \mid \mathbf{x}_\mathcal{R})$ and eliminate this possibility. We introduce the following new AEM as REAL-X:

$$\begin{aligned}\max_\beta \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\mathbb{E}_{\mathbf{s}_i\sim\text{Bernoulli}(f_\beta(\boldsymbol{x})_i)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}); \theta) - \lambda\|\boldsymbol{s}\|_0\right],\\ \max_\theta \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\mathbb{E}_{\boldsymbol{r}_i\sim\text{Bernoulli}(0.5)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{r}); \theta)\right].\end{aligned} \tag{4}$$

Implementation. REAL-X samples the selection variable for each feature $\mathbf{s}_i$ independently from a Bernoulli distribution. To control the number of features selected REAL-X sets $R(\mathbf{s}) = ||\mathbf{s}||_0$. To optimize this discrete process REAL-X uses score function, REBAR gradients [23] (see appendix B), where relaxed continuous selections are used within a control variate to lower the variance of the gradient estimates. The training procedures for REAL-X and q_{eval} are described in appendix C.

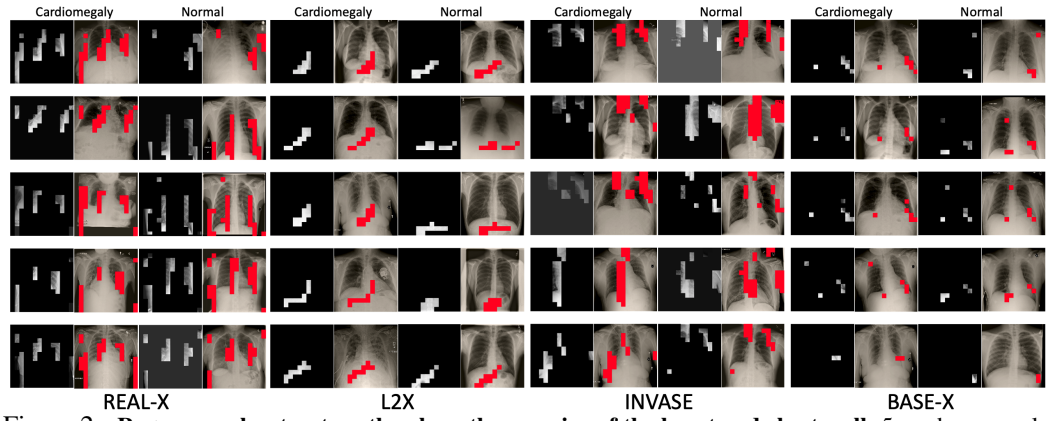

Figure 2: **REAL-X makes trustworthy along the margins of the heart and chest wall.** 5 random samples of Cardiomegaly and Normal Chest X-Rays are presented for each method, with the selections overlaid in red.

5 Explaining Cardiomegaly

Set-Up. A *simple* explanation of the data, one with fewer features selected, allows for greater human interpretability [8]. On imaging data this is likely to come at the cost of predictive accuracy. This trade-off can be encoded in the model selection process by tuning the hyper-parameter controlling the number of features selected and selecting the model that provides the simplest explanations while achieving sufficient predictive accuracy. We tune across $k = \{1, 5, 15, 50, 100\}$ for L2X and $\lambda = \{0.1, 1.0, 2.5, 5.0, 50.0\}$ for INVASE, REAL-X, and BASE-X, selecting the hyper-parameter that results in the simplest explanations such that the accuracy (ACC) is within 5% of a model trained on the full feature set. We introduce BASE-X as a JAM that mimics the gradient optimization procedure of REAL-X to ensure that the results we obtain on REAL-X are not due to changes in the optimization procedure. Post-hoc evaluation metrics, **eAUROC** and **eACC**, are then obtained by evaluating selections with q_{eval}.

Data. The NIH ChestX-ray8 Dataset[1] [24] contains $112,120$ chest X-rays from $30,805$ patients, each labeled with the presence of 8 diseases. We selected a small subset of $5,600$ X-rays labeled either *cardiomegaly* or *normal*, including all $2,776$ X-rays with cardiomegaly.

Model training. We used $5,000$, 300, and 300 images for training, validation, and testing respectively. UNet and DenseNet121 architectures were used for the selector and predictor models respectively providing 16×16 super-pixel selections. All methods were trained for 50 epochs using a learning rate of 10^{-4}.

Results. Table 1 shows that while each method makes selections that allow for high predictive performance (ACC$\geq 73.0\%$), REAL-X yields superior performance upon evaluation. Looking at selections of random Chest X-rays in fig. 2, we see that L2X, BASE-X and INVASE seem to make counterintuitive selections that omit many of the important pixels, resulting in a sharp decline in eACC.

Table 1: **real-x (REAL-X) yields superior post-hoc evaluation.**

Method	ACC	AUROC	eACC	eAUROC	$k\backslash\lambda$
All Features	78.0%	0.887	77.0%	0.884	–
REALX	75.0%	0.838	**70.3%**	**0.777**	2.5
L2X	75.0%	0.848	54.0%	0.581	10
INVASE	74.3%	0.819	52.3%	0.548	2.5
BASEX	74.3%	0.818	51.7%	0.595	2.5

Physician Evaluation. We asked two expert radiologists to rank each method based on the explanations provided. We randomly selected 50 chest X-rays from the test set and displayed each method's selections for each X-ray in a random order to the radiologists. For a given chest X-ray, the radiologists then evaluated which selections provided sufficient information to diagnose cardiomegaly and ranked the four options provided, allowing for ties. In table 2 we report the average rank each method achieved. We see that REAL-X consistently provides explanations that are meaningful to board-certified radiologists.

Table 2: **Average rankings by expert radiologists.**

REALX	L2X	INVASE	BASEX
1.08 (0.04)	3.57 (0.10)	2.85 (0.11)	2.29 (0.09)

[1] https://nihcc.app.box.com/v/ChestXray-NIHCC

6 Broader Impact

We began our discussion by acknowledging that model interpretations enable clinicians to trust model decisions, account for failure modes, and acquire new knowledge. However, all of these benefits hinge on the ability to trust the interpretations provided. If clinicians cannot trust model interpretations, they may loose faith in ML models or make decisions that negatively impact patient care. We examine a class of interpretability methods, amortized explanation methods (AEMs), that have the potential to seamlessly supplement the deployment of machine learning models in the clinic by offering computationally efficient, real-time interpretations. With this work, we hoped enable the community to trust AEMs, recognizing that prior AEMs may instead provide interpretations that encoding the prediction. We provide a way to check the interpretations provided by AEMs, and offer medical practitioners a new AEM, REAL-X, that addresses the issues with prior AEMs.

In practice, it may be unethical to provide unreliable explanations to clinicians or users in any field. We have addressed how this may affect patient care, but these issues extend to many other applications, such as criminal justice and finance, where providing poor model explanations can result in wrongful acquittals/convictions or unwise financial decisions. Additionally, for practitioners hoping to learn something new from the superhuman abilities of their models, incorrect explanations can shift understanding and have long term consequences.

While we focus our work on medical imaging, we believe that our work is broadly applicable to any field. We hope to both advance the ability of any machine learning practitioner to understand model decisions and bring attention to the need for trustworthy interpretations.

References

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515.

[2] Attia, Z. I., Kapa, S., Lopez-Jimenez, F., McKie, P. M., Ladewig, D. J., Satam, G., Pellikka, P. A., Enriquez-Sarano, M., Noseworthy, P. A., Munger, T. M., Asirvatham, S. J., Scott, C. G., Carter, R. E., and Friedman, P. A. (2019). Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. *Nature Medicine*, 25(1):70–74.

[3] Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2018). Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. Technical report.

[4] Cicero, M., Bilbily, A., Colak, E., Dowdell, T., Gray, B., Perampaladas, K., and Barfett, J. (2017). Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs. *Investigative radiology*, 52(5):281–287.

[5] Coudray, N., Ocampo, P. S., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyö, D., Moreira, A. L., Razavian, N., and Tsirigos, A. (2018). Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10):1559–1567.

[6] De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'Donoghue, B., Visentin, D., et al. (2018). Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350.

[7] Djuric, U., Zadeh, G., Aldape, K., and Diamandis, P. (2017). Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. *NPJ precision oncology*, 1(1):1–5.

[8] Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning.

[9] Geras, K. J., Wolfson, S., Shen, Y., Wu, N., Kim, S. G., Kim, E., Heacock, L., Parikh, U., Moy, L., and Cho, K. (2018). High-resolution breast cancer screening with multi-view deep convolutional neural networks.

[10] Ghorbani, A., Ouyang, D., Abid, A., He, B., Chen, J. H., Harrington, R. A., Liang, D. H., Ashley, E. A., and Zou, J. Y. (2020). Deep learning interpretation of echocardiograms. *npj Digital Medicine*, 3(1):10.

[11] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep

learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410.

[12] Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. (2019). A benchmark for interpretability methods in deep neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 9737–9748. Curran Associates, Inc.

[13] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax.

[14] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323.

[15] Lipton, Z. C. (2017). The mythos of model interpretability.

[16] Lundberg, S. M. and Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 4766–4775. Neural information processing systems foundation.

[17] Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

[18] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Model-Agnostic Interpretability of Machine Learning.

[19] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

[20] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

[21] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*.

[22] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*.

[23] Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., and Sohl-Dickstein, J. (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models.

[24] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[25] Yoon, J., Jordon, J., and van der Schaar, M. (2019). INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*.

[26] Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683.

[27] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8689 LNCS, pages 818–833. Springer Verlag.

[28] Zhou, J. and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12(10):931–934.

[29] Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. (2017). Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

A L2X and INVASE are JAMs

INVASE is a JAM that models the selector variable $\mathbf{s}$ using independent Bernoulli distributions denoted $\mathcal{B}$ whose probabilities are given by a function f of the features. It sets $R(\mathbf{s}) = \ell_0(\mathbf{s})$ to enforce sparse feature selections and uses the following masking function:

$$m(\boldsymbol{x}^{(i)}, \boldsymbol{s}^{(i)})_j = \begin{cases} \boldsymbol{x}_j^{(i)} & \text{if } \boldsymbol{s}_j^{(i)} = 1 \\ [\texttt{mask}] & \text{if } \boldsymbol{s}_j^{(i)} = 0 \end{cases}. \tag{5}$$

INVASE also uses $q_{\text{control}}(\boldsymbol{y} \mid \boldsymbol{x}; \phi)$ as a control variate within the objective to reduce the variance of the score function gradients during optimization. The INVASE objective for learning θ and β is

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\mathbb{E}_{\boldsymbol{s}_j\sim\text{Bernoulli}(f_\beta(\boldsymbol{x})_j)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}); \theta) - \log q_{\text{control}}(\boldsymbol{y} \mid \boldsymbol{x}; \phi) - \lambda\|\mathbf{s}\|_0\right]$$

The use of q_{control} does not alter INVASE's objective with respect to the selector or predictor model, so it fits into the form of eq. (2).

L2X is a JAM that uses k independent samples from a Concrete distribution [17, 13] to define the selector model in order to make use of reparameterization gradients during optimization. In L2X, $\mathbf{s}$ is sampled from $q_{\text{sel}}(\mathbf{s} \mid \boldsymbol{x}; \beta)$, where

$$\boldsymbol{c}_j \sim \text{Concrete}(f_\beta(\boldsymbol{x})), \quad C = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k] \in \mathbb{R}^{D\times k}, \quad \boldsymbol{s}_i = \max_{1\le j\le k} C_{ij}.$$

Selection with the mask function is accomplished with multiplication: $m(\mathbf{x}, \mathbf{s}) = \mathbf{x} \odot \mathbf{s}$. Sparse selections are enforced by limiting the number of samples taken from a Concrete distribution, assigning a hard bound k on the number of features selected. This results in the following objective for learning θ and β:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\mathbb{E}_{s\sim q_{\text{sel}}(\mathbf{s} \mid \boldsymbol{x};\beta)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid \boldsymbol{x} \odot \boldsymbol{s}; \theta)\right], \tag{6}$$

assuming the form of eq. (2).

B Applying REBAR Gradient Estimation to REAL-X

Computing the gradient of an expectation of a function with respect to the parameters of a discrete distribution requires calculating score function gradients. Score function gradients are high variance. To reduce the variance, control variates are used within the objective. REBAR gradient calculation involves using a highly correlated control variate that approximates the discrete distribution with its continuous relaxation.

The REAL-X procedure involves

$$\max_\beta \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\mathbb{E}_{\mathbf{s}_i\sim\mathcal{B}(f_\beta(\boldsymbol{x})_i)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}); \theta) - \lambda\|\boldsymbol{s}\|_0\right].$$

This is accomplished through stochastic gradient ascent by taking

$$\nabla_\beta \mathbb{E}_{\mathbf{s}_i\sim\mathcal{B}(f_\beta(\boldsymbol{x})_i)}\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}); \theta) - \lambda\|\boldsymbol{s}\|_0\right], \tag{7}$$

which requires score function gradient estimation.

Let $\mathbf{s}$ be a discrete random variable, $\mathcal{L} = \mathbb{E}_{\mathbf{s}\sim q_\beta}[h(\mathbf{s})]$, and $\mathbb{E}[\hat{g}_\beta] = \nabla_\beta\mathcal{L}$, the REBAR gradient estimator [23] computes $\hat{g}_\beta$. Then, letting $\mathbf{z}$ be a continous relaxation of $\mathbf{s}$, REBAR estimates the gradient as

$$\hat{g}_\beta = [h(\boldsymbol{s}) - h(\tilde{\boldsymbol{z}})]\nabla_\beta \log q_\beta(\boldsymbol{s}) - \nabla_\beta h(\tilde{\boldsymbol{z}}) + \nabla_\beta h(\boldsymbol{z}),$$
$$\text{where} \quad \boldsymbol{s} = B(\boldsymbol{z}),\ \boldsymbol{z} \sim q_\beta(\mathbf{z}),\ \tilde{\boldsymbol{z}} \sim q_\beta(\mathbf{z}|\boldsymbol{s}).$$

To estimate eq. (7) using REBAR, REAL-X sets

$$h(\boldsymbol{s}) = \log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x},\ \boldsymbol{s}\ ); \theta).$$

Here, $\mathbf{s}$ is Bernoulli distributed and REAL-X sets $\mathbf{z}$ to be distributed as the binary equivalent of the Concrete distribution [17, 13], which we refer to as the *RelaxedBernoulli* distribution. $\mathbf{s}$, $\mathbf{z}$, and $\tilde{\mathbf{z}}$ are

sampled as described by Tucker et al. [23] such that

$$p_i = f_\beta(\boldsymbol{x})_i,$$
$$\boldsymbol{s}_i = B(\boldsymbol{z}_i) = \mathbb{1}(\boldsymbol{z}_i > 0), \tag{8}$$
$$\boldsymbol{z}_i \sim q_\beta(\mathbf{z} \mid \boldsymbol{x}) = \textit{RelaxedBernoulli}(p_i; \tau = 0.1), \tag{9}$$
$$\tilde{\mathbf{z}}_i \sim q_\beta(\mathbf{z} \mid \boldsymbol{x}, \boldsymbol{s}) = \frac{1}{0.1}\left(\log\frac{p_i}{1-p_i} + \log\frac{\boldsymbol{v}'}{1-\boldsymbol{v}'}\right), \tag{10}$$
$$\text{where } \boldsymbol{v} \sim \text{Unif}(0,1) \text{ and } \boldsymbol{v}' = \begin{cases} \mathbf{v}(1-p_i) & \text{if } \boldsymbol{s}_i = 0 \\ \mathbf{v}p_i + (1-p_i) & \text{if } \boldsymbol{s}_i = 1 \end{cases}.$$

Then to estimate eq. (7) notice that

$$\nabla_\beta \mathbb{E}_{\mathbf{s}_i \sim \mathcal{B}(f_\beta(\boldsymbol{x})_i)}\left[\lambda \|\boldsymbol{s}\|_0\right] = \lambda \nabla_\beta f_\beta(\mathbf{x}).$$

REAL-X, therefore, estimates eq. (7) by calculating $\hat{g}_\beta$ as

$$\begin{aligned}\hat{g}_\beta = &\left[\log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s})) - \log q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \tilde{\boldsymbol{z}}))\right] \nabla_\beta \log q_{\text{sel}}(\boldsymbol{s} \mid \boldsymbol{x}; \beta) - \lambda \nabla_\beta f_\beta(\boldsymbol{x}) \\ &- \nabla_\beta q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \tilde{\boldsymbol{z}})) + \nabla_\beta q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{z}))\end{aligned} \tag{11}$$

C Algorithms

C.1 REAL-X algorithm

Algorithm 1 REAL-X Algorithm

Input: $\mathcal{D} := (\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathbb{R}^{N \times D}$, feature matrix; $\boldsymbol{y} \in \mathbb{R}^N$, labels
Output: $q_{\text{sel}}(\mathbf{s} \mid \boldsymbol{x})$, function that returns feature selections given an instance of $\mathbf{x}$
Select: λ, regularization constant; α, learning rate; M, mini-batch size, T, training-steps
for $1, ..., T$ **do**
  Randomly sample mini-batch of size M, $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})_{i=1}^M \sim \mathcal{D}$
  for $i = 1, ..., M$ **do**
    Sample Selections:
      $\boldsymbol{r}^{(i)} \sim \text{Bernoulli}(0.5)$
      Sample $\boldsymbol{s}^{(i)}$, $\boldsymbol{z}^{(i)}$, and $\tilde{\boldsymbol{z}}^{(i)}$ as in eqs. (8) to (10)
  end
  Optimize Models:
    $\theta = \theta + \alpha \nabla_\theta \left[\frac{1}{M}\sum_{i=1}^M \log q_{\text{pred}}(\boldsymbol{y}^{(i)} | m(\boldsymbol{x}^{(i)}, \boldsymbol{r}^{(i)}; \theta)\right]$
    $\beta = \beta + \alpha \frac{1}{M}\sum_{i=1}^M \hat{g}_\beta$, where $\hat{g}_\beta$ is calculated as in eq. (11)
end

C.2 Evaluation Algorithm

Algorithm 2 Algorithm to Train Evaluator Model q_{eval}

Input: $\mathcal{D} := (\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathbb{R}^{N \times D}$, feature matrix; $\boldsymbol{y} \in \mathbb{R}^N$, labels
Output: $q_{\text{eval}}(\mathbf{y} \mid m(\boldsymbol{x}, \cdot); \eta)$, function that returns the probability of the target given a subset of features.
Select: α, learning rate; M, mini-batch size
while *Converge* **do**
  Randomly sample mini-batch of size M, $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})_{i=1}^M \sim \mathcal{D}$
  for $i = 1, ..., M$ **do**
    Sample Selections:
      $\boldsymbol{r}^{(i)} \sim \text{Bernoulli}(0.5)$
  end
  Optimize:
    $\eta = \eta + \alpha \nabla_\eta \left[\frac{1}{M}\sum_{i=1}^M \log q_{\text{eval}}(\boldsymbol{y}^{(i)} | m(\boldsymbol{x}^{(i)}, \boldsymbol{r}^{(i)}); \eta)\right]$
end

D Additional Lemmas

Lemma 2. *Let* $\mathbf{x} \in \mathbb{R}^D$*, target* $\mathbf{y} \in \{1, ..., K\}$*, and* Δ *be a set of* K *dimensional probability vectors, then for* $J = \arg\min_j \sum_{i=0}^{j} \binom{D}{i} \geq |\Delta|$ *and* $\mathbf{x} \sim F$*, there exists a* q_{sel} *and* q_{pred}*, where* $\{q_{pred}(y = k \mid m(\mathbf{x}, \mathbf{s}))\}_{k=1}^{K} = \delta(\mathbf{x}) \in \Delta$ *and* $E[||\mathbf{s}||_0] \leq J$*.*

E Proofs

E.1 Proof of Lemma 1

Lemma 1. *Let* $\mathbf{x} \in \mathbb{R}^D$ *and target* $\mathbf{y} \in \{1, ..., K\}$*. If* $\mathbf{y}$ *is a deterministic function of* $\mathbf{x}$ *and* $K \leq D$*, then* JAMs *with monotone increase regularizers* R *will select at most one feature at optimality.*

As mentioned in section 3, the lemma considers the masking function from eq. (5) and on independent Bernoulli selector variables $\mathbf{s}_j \sim \text{Bernoulli}(f_\beta(\mathbf{x})_j)$.

$\mathbf{s} \in \mathbb{R}^D$ is binary and, therefore, has the capacity to transmit D bits of information. Given that $\mathbf{y} \in \{1, ..., K\}$ is a deterministic function of $\mathbf{x} \in \mathbb{R}^D$, the true distribution is $F(\mathbf{y} \mid \boldsymbol{x}) \in \{0, 1\}$ for each of the K realizations of $\mathbf{y}$. Therefore, $m(\boldsymbol{x}, \boldsymbol{s})$ must pass at least $\log_2 K$ bits of information to the predictor model $q_{\text{pred}}(\mathbf{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))$.

With m of the form *eq.* (5), this information content can come from $\boldsymbol{s}$. $\boldsymbol{s}$ has a capacity of $\log_2 \left(\sum_{i=1}^{n} \binom{D}{i} \right)$ bits when restricted to realizations of $\boldsymbol{s} \sim q_{\text{sel}}$ with at most n non-zero elements. The maximal number of non-zero elements J in any given realization of $\mathbf{s}$ required to minimally transmit $\log_2 K$ bits of information with $\boldsymbol{s}$ can be expressed as

$$J = \arg\min_j \sum_{i=0}^{j} \binom{D}{i} \geq K.$$

Given $K \leq D$, the maximal number of selections required is given by $J = 1$, where $\binom{D}{1} = D \geq K$. Therefore there exists a q_{pred} and q_{sel} such that $\mathbb{E}[q_{\text{pred}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))] = \mathbb{E}[F(\mathbf{y} \mid \mathbf{x})]$ and $\mathbb{E}[\|\mathbf{s}\|_0] \leq 1$. For monotone increasing regularizer R, any solution that selects more than a single feature will have a lower JAM objective. Therefore, at optimally, JAMs will select at most a single feature.

E.2 Proof of Lemma 2

Lemma 2. *Let* $\mathbf{x} \in \mathbb{R}^D$*, target* $\mathbf{y} \in \{1, ..., K\}$*, and* Δ *be a set of* K *dimensional probability vectors, then for* $J = \arg\min_j \sum_{i=0}^{j} \binom{D}{i} \geq |\Delta|$ *and* $\mathbf{x} \sim F$*, there exists a* q_{sel} *and* q_{pred}*, where* $\{q_{pred}(y = k \mid m(\mathbf{x}, \mathbf{s}))\}_{k=1}^{K} = \delta(\mathbf{x}) \in \Delta$ *and* $E[||\mathbf{s}||_0] \leq J$*.*

This proof follows from the proof in appendix E.1. Given $\mathbf{x} \in \mathbb{R}^D$ and target $\mathbf{y} \in \{1, ..., K\}$, there exists a distribution $q_{\text{pred}}(\mathbf{y} \mid m(\mathbf{x}, \mathbf{s}))$ such that each realization of $\mathbf{s} \in \{0, 1\}^D$ has a bijective mapping to a unique probability vector obtained as $\{q_{\text{pred}}(y = k \mid m(\mathbf{x}, \mathbf{s}))\}_{k=1}^{K} \in \mathbb{R}^K$.

As stated in the proof of lemma 1 $\mathbf{s}$ has a capacity of $\log_2 \left(\sum_{i=1}^{n} \binom{D}{i} \right)$ bits when restricted to realizations of $\mathbf{s} \sim q_{\text{sel}}$ with at most n non-zero elements. Given a set of K dimensional probability vectors Δ, the maximal number of non-zero selections in $\mathbf{s}$ required to produce at least $|\Delta|$ unique realizations of $\mathbf{s}$, denoted by J, can be expressed as

$$J = \arg\min_j \sum_{i=0}^{j} \binom{D}{i} \geq |\Delta|.$$

Then there exists a q_{pred} and q_{sel} such that there are at least $|\Delta|$ unique probability vectors $\{q_{\text{pred}}(y = k \mid m(\mathbf{x}, \mathbf{s}))\}_{k=1}^{K} = \delta(\mathbf{x}) \in R^K$ where $\delta(\mathbf{x}) \in \Delta$ and the average number of features selected $E[||\mathbf{s}||_0] \leq J$.

F Optimality of the Evaluator Model

The evaluator model q_{eval} is learned such that eq. (3) is maximized as follows:

$$\max_{\eta} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\mathbb{E}_{\boldsymbol{r}_i\sim\text{Bernoulli}(0.5)}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid m(\boldsymbol{x},\boldsymbol{r});\eta)\right].$$

We aim to show that this expectation is maximal when $q_{\text{eval}}(\boldsymbol{y}\mid m(\boldsymbol{x},\boldsymbol{r})) = F(\boldsymbol{y}\mid \boldsymbol{x}_{\mathcal{R}})$ for any sample of $\mathbf{r}$ identifying the corresponding subset of features $\mathcal{R}$ in the input $\boldsymbol{x}_{\mathcal{R}}$.

The expectations can be rewritten as

$$\max_{\eta} \mathbb{E}_{\boldsymbol{r}_i\sim\text{Bernoulli}(0.5)}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\mid \boldsymbol{r}\sim F}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid m(\boldsymbol{x},\boldsymbol{r});\eta)\right].$$

Let the power set over feature selections $\mathcal{P}_{\boldsymbol{r}} = \{\boldsymbol{r}\subset\{0,1\}^D\}$ and equivalently for the corresponding feature subsets $\mathcal{P}_R = \{\mathcal{R}\subset 2^D\}$. Given $\boldsymbol{r}_i\sim\text{Bernoulli}(0.5)$, the probability

$$p(\boldsymbol{r}) = \frac{1}{|\mathcal{P}_{\boldsymbol{r}}|} = \frac{1}{|\mathcal{P}_R|}.$$

Recognizing that $\mathbf{x},\mathbf{y}\perp\mathbf{r}$, the expectation over $\mathbf{r}$ can be expanded as

$$\max_{\eta}\sum_{\boldsymbol{r}\in\mathcal{P}_{\boldsymbol{r}}}\frac{1}{|\mathcal{P}_{\boldsymbol{r}}|}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid m(\boldsymbol{x},\boldsymbol{r});\eta)\right].$$

Here, the expectation is with respect to a given $\boldsymbol{r}$ in the power set $\mathcal{P}_{\boldsymbol{r}}$. In this case, neither $\boldsymbol{r}$ nor the subset of features masked by $m(\boldsymbol{x},\boldsymbol{r})$ provide any information about the target. Therefore, the likelihood is calculated with respect to the corresponding fixed subset $\mathcal{R}$ as

$$\max_{\eta}\sum_{\mathcal{R}\in\mathcal{P}_R}\frac{1}{|\mathcal{P}_R|}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid \boldsymbol{x}_{\mathcal{R}};\eta)\right].$$

A finite sum is maximized when each individual element in the sum is maximized, therefore it suffices to find

$$\max_{\eta}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid \boldsymbol{x}_{\mathcal{R}};\eta)\right]\quad\forall\mathcal{R}\in\mathcal{P}_R$$

Let $q_{\text{eval}} := \{f_{\mathcal{R}}(\,\cdot\,;\eta_{\mathcal{R}})\}_{\mathcal{R}\in\mathcal{P}_R}$, such that when given $\boldsymbol{r}$ as an input for the corresponding $\mathcal{R}$, $f_{\mathcal{R}}(\,\cdot\,;\eta_{\mathcal{R}})$ is used to generate the target. The key point here is that the subset $\mathcal{R}$ provided to the model as $\boldsymbol{r}$ can uniquely identify which $f_{\mathcal{R}}$ generates the target. Then, for any given R, each expectation is maximized when the corresponding $f_{\mathcal{R}}$ is equal to the true data generating distribution given by

$$\begin{aligned}\max_{\eta}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim F}\left[\log q_{\text{eval}}(\boldsymbol{y}\mid \boldsymbol{x}_{\mathcal{R}};\eta)\right] &= \max_{\eta_{\mathcal{R}}}\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\log f_{\mathcal{R}}(\mathbf{y}\mid \mathbf{x}_{\mathcal{R}};\eta_{\mathcal{R}})\right]\\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log F(\mathbf{y}\mid \mathbf{x}_R)]\quad\forall\mathcal{R}\in\mathcal{P}_R.\end{aligned}$$