Supplementary Materials

A 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 often have high variance. To reduce this 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

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

This is accomplished through stochastic gradient ascent by taking

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

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 (Tucker et al., 2017) computes $\hat{g}_\beta$. Then, letting $\mathbf{z}$ be a continous relaxation of $\mathbf{s}$, REBAR estimates the gradient as

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

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 (Maddison et al., 2016; Jang et al., 2017), which we refer to as the *RelaxedBernoulli* distribution. $\mathbf{s}$, $\mathbf{z}$, and $\tilde{\mathbf{z}}$ are sampled as described by Tucker et al. (2017) such that

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

Then to estimate eq. (7) notice that

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

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

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

B Algorithms

B.1 Evaluation Algorithm

Algorithm 2 Algorithm to Train Evaluator Model $q_{\text{eval-x}}$

```
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: $q_{\text{eval-x}}(\mathbf{y} \mid m(\boldsymbol{x}, \cdot); \eta)$, function that returns the probability of the target given a subset of features.
Select: $\alpha$, 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:
        $\eta = \eta + \alpha \nabla_\eta \left[ \frac{1}{M} \sum_{i=1}^{M} \log q_{\text{eval-x}}(\boldsymbol{y}^{(i)} | m(\boldsymbol{x}^{(i)}, \boldsymbol{r}^{(i)}); \eta) \right]$
end
```

C Lemmas

Lemma 3. *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 $E[||\mathbf{s}||_0] \leq J$.*

D Proofs

D.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 mentioning in section 3, the lemma considers the masking function from eq. (2) 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. (2)*, 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.

D.2 Proof of Lemma 3

Lemma 3. *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 D.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 $E[||\mathbf{s}||_0] \leq J$.

D.3 Proof of Lemma 2

Lemma 2. *Assume that the true $F(\mathbf{y} \mid \mathbf{x})$ is computed as a tree, where the leaves ℓ_i are the conditional distributions $F_i(\mathbf{y} \mid \mathbf{x}_{\mathcal{S}_i})$ of $\mathbf{y}$ given distinct subsets of features $\mathcal{S}_i$ in $\mathbf{x}$. Given a monotone increasing regularizer $R(|\mathcal{S}_i|)$, the preferred maximizer of the* JAM *objective excludes control flow features.*

The main intuition behind the proof of lemma 2 is as follows. The JAM objective results in a prediction model that does not require the control flow features to achieve optimal performance. As a result of the monotone increasing regularizer R, which assigns a cost for selecting each additional feature, the JAM objective omits control flow features. We now prove this idea formally.

The tree is structured such that each leaf ℓ_i in the tree has a corresponding conditional distribution $F_i(\mathbf{y} \mid \mathbf{x}_{\mathcal{S}_i})$ parameterized by a set of features $\mathcal{S}_i$ such that $\forall j \neq i, \mathcal{S}_i \neq \mathcal{S}_j$. Let the features found along the the path from the root of the tree to the leaf, including those found at the leaf, be defined as $\mathcal{T}_i$ for each leaf ℓ_i. Those features that are not in the leaf and only appear in the non-leaf nodes of the tree are the control flow features defined as $\mathcal{C}_i := \mathcal{T}_i \backslash \mathcal{S}_i$. For any input $\boldsymbol{x}$, let $\mathcal{T}(\boldsymbol{x})$ be the features found along the path used in generating the response for that $\boldsymbol{x}$ and define the control flow features along the path as $\mathcal{C}(\boldsymbol{x})$ and set of leaf features $\mathcal{S}(\boldsymbol{x})$.

Consider the following cases where $q_{\text{sel}}(\mathbf{s} \mid \boldsymbol{x})$ selects the jth feature with probability

$$\begin{cases} q_{\text{sel1}}(\mathbf{s}_j \mid \boldsymbol{x}) = \mathbb{1}[j \in \mathcal{T}(\boldsymbol{x})] = \mathbb{1}[j \in \{\mathcal{C}(\boldsymbol{x}) \cup \mathcal{S}(\boldsymbol{x})\}] & \text{(Case 1)} \\ q_{\text{sel1}}(\mathbf{s}_j \mid \boldsymbol{x}) = \mathbb{1}[j \in \mathcal{S}(\boldsymbol{x})] & \text{(Case 2)} \end{cases},$$

where q_{sel1} and q_{sel1} denotes the q_{sel} for case 1 and case 2 respectively. $q_{\text{pred1}}(\mathbf{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))$ and $q_{\text{pred2}}(\mathbf{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))$ are defined in the corresponding manner.

In case 1, the predictor model q_{pred1} receives all the relevant features from q_{sel1}, such that $\mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} \mathbb{E}_{\boldsymbol{s} \sim q_{\text{sel1}}(\mathbf{s} \mid \boldsymbol{x})} [\log q_{\text{pred1}}(\mathbf{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))]$ can predict as well as possible.

In case 2, however, the predictor model $q_{\text{pred2}}(\mathbf{y} \mid m(\boldsymbol{x}, \boldsymbol{s}))$ does not receive the full set of relevant features from q_{sel2}; it only receives the leaf features. Since the leaf features are unique across leaves, the selections indicated by $\boldsymbol{s}$ provides enough information for the predictor model to consistently learn the correct data generating leaf conditional $F(\mathbf{y} \mid \boldsymbol{x}_{\mathcal{S}(\boldsymbol{x})})$, meaning that it can predict as well as possible.

Assuming the models maximize the JAM objective, in both cases q_{pred} together with q_{sel} correctly model $F(\mathbf{y} \mid \mathbf{x})$. Plugging this information into the JAM objective in eq. (3) yields the following:

$$\begin{aligned} \mathcal{L}_{\text{Case 1}} &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} \mathbb{E}_{\mathcal{T}(\boldsymbol{x}) \sim q_{\text{sel1}}(\mathbf{s} \mid \boldsymbol{x})} [\log q_{\text{pred1}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \mathcal{T}(\boldsymbol{x}))) - \lambda R(|\mathcal{T}(\boldsymbol{x})|)] \\ &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} [\log F(\boldsymbol{y} \mid \boldsymbol{x})] - \lambda \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} \mathbb{E}_{\mathcal{T}(\boldsymbol{x}) \sim q_{\text{sel1}}(\mathbf{s} \mid \boldsymbol{x})} [R(|\mathcal{T}(\boldsymbol{x})|)], \\ \mathcal{L}_{\text{Case 2}} &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} \mathbb{E}_{\mathcal{S}(\boldsymbol{x}) \sim q_{\text{sel2}}(\mathbf{s} \mid \boldsymbol{x})} [\log q_{\text{pred2}}(\boldsymbol{y} \mid m(\boldsymbol{x}, \mathcal{S}(\boldsymbol{x}))) - \lambda R(|\mathcal{S}(\boldsymbol{x})|)] \\ &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} [\log F(\boldsymbol{y} \mid \boldsymbol{x})] - \lambda \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim F} \mathbb{E}_{\mathcal{S}(\boldsymbol{x}) \sim q_{\text{sel2}}(\mathbf{s} \mid \boldsymbol{x})} [R(|\mathcal{S}(\boldsymbol{x})|)]. \end{aligned}$$

Given that $R(.)$ is monotone increasing, the following inequality holds:

$$\mathcal{L}_{\text{Case 2}} \geq \mathcal{L}_{\text{Case 1}}.$$

For any $\lambda > 0$ where control flow features are involved in the data generating process, that is $\mathcal{C}_i \neq \emptyset$ for some i, this inequality is strict. Therefore, the solution that omits control flow features (Case 2) will have a higher objective value, which we describe as the preferred maximizer of the JAM objective. Thus, at optimality, control flow features will not be selected under the JAM objective with a monotone increasing regularizer.

E Optimality of the Evaluator Model

The evaluator model $q_{\text{eval-x}}$ is learned such that eq. (5) is maximized as follows:

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

We aim to show that this expectation is maximal when $q_{\text{eval-x}}(\boldsymbol{y} \,|\, m(\boldsymbol{x},\boldsymbol{r})) = F(\boldsymbol{y} \,|\, \boldsymbol{x}_{\mathcal{R}})$ 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

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

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

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

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

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

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

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

Let $q_{\text{eval-x}} := \{f_{\mathcal{R}}(\,\cdot\,; \eta_{\mathcal{R}})\}_{\mathcal{R}\in\mathcal{P}_R}$, 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

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

F Additional Experiments

F.1 EVAL-X vs. Models Explicitly Trained For Each Feature Subset.

In this experiment, we evaluate EVAL-X. EVAL-X approximates $F(\mathbf{y} \mid \mathbf{x}_{\mathcal{R}})$ for any subset of features $\mathcal{R}$, by training on randomly sampled subsets of the input $\mathbf{x}$. While, at optimally the training procedure for EVAL-X returns a model of $F(\mathbf{y} \mid \mathbf{x}_{\mathcal{R}})$, it may be difficult to approximate the distribution $F(\mathbf{y} \mid \mathbf{x}_{\mathcal{R}})$ for every possible subset of features. We therefore trained separate models for each unique subset of features on our synthetic dataset described in section 6.3. Each of the datasets contain 11 input features with 2048 distinct feature subsets. For each dataset we trained 2048 distinct models on each feature subset. We then evaluated the selections made by each AEM on this collection of models and on EVAL-X. We compared the AUROC returned by EVAL-X (eAUROC) to those returned by the collection of models (cAUROC) in table 5. While EVAL-X returns underestimates relative to the collection of models, the difference is small and the trend amongst methods is conserved.

Table 5: **REAL-X yields superior post-hoc evaluation on a collection of each models for each feature subset.**

	S1		S2		S3	
Method	eAUROC	cAUROC	eAUROC	cAUROC	eAUROC	cAUROC
REAL-X	0.774	**0.798**	0.804	**0.807**	0.873	**0.876**
L2X	0.742	0.759	0.771	0.776	0.848	0.849
INVASE	0.740	0.767	0.783	0.788	0.868	0.870
BASE-X	0.762	0.773	0.773	0.777	0.867	0.870

F.2 Training the Predictor Model First.

We compared a REAL-X approach where q_{pred} is first fully optimized, then q_{sel} is optimized in a stepwise manor (REAL-X-STEP) to the approach outlined in algorithm 1, where both q_{sel} and q_{pred} are optimized simultaneous with each mini-batch. The AUROCs returned by EVAL-X for the synthetic datasets described in section 6.3 are presented in table 6. Both approaches perform similarly.

Table 6: **REAL-X and REAL-X-STEP perform similarly.**

	S1		S2		S3	
Metric	REAL-X	REAL-X-STEP	REAL-X	REAL-X-STEP	REAL-X	REAL-X-STEP
eAUROC	0.774	0.778	0.804	0.801	0.873	0.872