Appendix to FasterRisk: Fast and Accurate Interpretable Risk Scores

Table of Contents

A Additional Algorithms **16**
- A.1 Expand Support by One More Feature . . . 16
- A.2 Collect Sparse Diverse Pool . . . 16
- A.3 Round Continuous Coefficients to Integers . . . 17

B Comments on Proof of Chevaleyre *et al.* **17**

C Theoretical Upper Bound for the Rounding Method, Algorithm 6 **18**

D Experimental Setup **22**
- D.1 Dataset Information . . . 22
- D.2 Computing Platform . . . 22
- D.3 Baselines . . . 22
- D.4 Hyperparameters Specification . . . 24

E Additional Experimental Results **25**
- E.1 Additional Results on Solution Quality . . . 25
- E.2 Additional Results on Direct Comparison with RiskSLIM . . . 26
- E.3 Additional Results on Running Time . . . 26
- E.4 Ablation Study of the Proposed Techniques . . . 27
- E.5 Training Losses of FasterRisk vs. RiskSLIM . . . 30
- E.6 Comparison of SparseBeamLR with OMP and fastSparse . . . 31
- E.7 Comparison of FasterRisk with OMP (or fastSparse) + Sequential Rounding . . . 35
- E.8 Running RiskSLIM Longer . . . 38
- E.9 Calibration Curves . . . 43
- E.10 Hyperparameter Perturbation Study . . . 46
- E.11 Comparison with Baseline AutoScore . . . 58

F Additional Risk Score Models **61**
- F.1 Risk Score Models with Different Sizes . . . 61
- F.2 Risk Score Models from the Pool of Solutions . . . 69

G Model Reduction **81**
- G.1 Reducing Models to Relatively Prime Coefficients . . . 81
- G.2 Transforming Features for Better Interpretability . . . 82

H Discussion of Limitations **83**

A Additional Algorithms

A.1 Expand Support by One More Feature

Algorithm 4 ExpandSuppBy1

Input: Dataset $\mathcal{D}$, coefficient constraint C, and beam search size B, current coefficient vector $(\boldsymbol{w}, w_0)$, and a set of found supports $\mathcal{F}$.
Output: a collection of solutions $\mathcal{W} = \{(\boldsymbol{u}^t, u_0^t)\}$ with $\|\boldsymbol{u}^t\|_0 = \|\boldsymbol{w}\|_0 + 1$, $\|\boldsymbol{u}^t\|_\infty \leq C$ for $\forall t$. All of these solutions include the support of $(\boldsymbol{w}, w_0)$ plus one more feature. None of the solutions have the same support as any element of $\mathcal{F}$, meaning we do not discover the same support set multiple times. We also output the updated $\mathcal{F}$.

Let $\mathcal{S}^c \leftarrow \{j \mid w_j = 0\}$ ▷*Non-support of the given solution*
$\boldsymbol{w}' \leftarrow \mathbf{0}$
for $p = 1, ..., 10$ **do** ▷*10 steps of parallel coordinate descent with projection*
    $w'_j \leftarrow w'_j - \nabla_j L(\boldsymbol{w} + w'_j \boldsymbol{e}_j, w_0)/l_j$ for $\forall j \in \mathcal{S}^c$ ▷*l_j is the smallest Lipschitz constant on coordinate j with $L(\boldsymbol{w} + w'_j \boldsymbol{e}_j + d\boldsymbol{e}_j) - L(\boldsymbol{w} + w'_j \boldsymbol{e}_j) \leq l_j d$ for any $d \in \mathbb{R}$.*
    $w'_j \leftarrow \text{Clip}(w'_j, -C, C)$ for $\forall j \in \mathcal{S}^c$ ▷*Clip*$(x, a, b) = \max(a, \min(x, b))$
end for
Pick the B coords (j's) in $\mathcal{S}^c$ with smallest logistic loss $L(\mathcal{D}, \boldsymbol{w} + \boldsymbol{e}_j w'_j, w_0)$, call this set $\mathcal{J}'$. ▷*We will use these supports, which include the support of $\boldsymbol{w}$ plus one more.*
$\mathcal{W} \leftarrow \emptyset$
for $j \in \mathcal{J}'$ **do** ▷*Optimize on the top B coordinates*
    If $\text{supp}(\boldsymbol{w} + \boldsymbol{e}_j w'_j) \in \mathcal{F}$, continue. ▷*We've already seen this support, so skip.*
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{supp(\boldsymbol{w} + \boldsymbol{e}_j w'_j)\}$. ▷*Add new support to $\mathcal{F}$.*
    $(\boldsymbol{w}'', w''_0) \in \text{argmin}_{\boldsymbol{u}, u_0} L(\mathcal{D}, \boldsymbol{u}, u_0)$ with $\text{supp}(\boldsymbol{u}) = \text{supp}(\boldsymbol{w} + \boldsymbol{e}_j w'_j)$ and $\|\boldsymbol{u}\|_\infty \leq C$. ▷*Fine tune on the newly expanded support using 100×|support| coordinate descent steps and clip operation, or until convergence; use $(\boldsymbol{w} + \boldsymbol{e}_j w'_j, w_0)$ as a warm start for computational efficiency*
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{(\boldsymbol{w}'', w''_0)\}$
end for
Return $\mathcal{W}$ and $\mathcal{F}$.

A.2 Collect Sparse Diverse Pool

Algorithm 5 CollectSparseDiversePool

Input: Dataset $\mathcal{D}$, a coefficient vector $(\boldsymbol{w}, w_0)$, an optimality gap tolerance ϵ, and the number of attempts T.
Output: a set $\mathcal{S}$ containing good sparse continuous solutions.

$\mathcal{S} \leftarrow \{(\boldsymbol{w}, w_0)\}$ ▷*Initialize the sparse level set*
$L^* \leftarrow L(\mathcal{D}, \boldsymbol{w}, w_0)$ ▷*Get the current loss*
$\mathcal{J} \leftarrow \{j \mid w_j \neq 0\}$ ▷*Get the current support*
for $j_- \in \mathcal{J}$ **do** ▷*Remove a feature in the support*
    Pick the T coords (j_+'s) in $[1, ..., p] \setminus \mathcal{J}$ with the biggest magnitudes of partial derivative $\nabla_{j^+} L(\mathcal{D}, \boldsymbol{w} - \boldsymbol{e}_{j_-} w'_j, w_0)$, call this set $\mathcal{J}_+$.
    for $j_+ \in \mathcal{J}_+$ **do** ▷*Put a new feature into the support*
        $(\boldsymbol{w}'', w''_0) \in \text{argmin}_{\boldsymbol{w}', w'_0} L(\mathcal{D}, \boldsymbol{w}', w'_0)$ where $w'_j = 0$ if $j \in [1, ..., p] \setminus \mathcal{J} \cup \{j_-\}$ ▷*Fit on the new support. Problem is convex. We use coordinate descent for this.*
        $L_{\text{swap}} \leftarrow L(\mathcal{D}, \boldsymbol{w}'', w''_0)$ ▷*Loss of newly formed and optimized coefficient vector*
        if $L_{\text{swap}} \leq (1 + \epsilon) L^*$ **then** ▷*If its loss is good enough, include it in $\mathcal{S}$*
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\boldsymbol{w}'', w''_0)\}$ ▷*Expand the sparse level set if loss is within the gap*
        end if
    end for
end for
return $\mathcal{S}$

A.3 Round Continuous Coefficients to Integers

Algorithm 6 AuxiliaryLossRounding

Input: Dataset $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^n$, a sparse continuous solution $(\boldsymbol{w}, w_0)$, where $\boldsymbol{w} \in \mathbb{R}^p, w_0 \in \mathbb{R}$.
Output: an integer-valued solution $(\boldsymbol{w}^+, w_0^+)$, where $\boldsymbol{w}^+ \in \mathbb{Z}^p, w_0^+ \in \mathbb{Z}$.

$\boldsymbol{w}^c \leftarrow [w_0, \boldsymbol{w}]$, and $\boldsymbol{x}_i \leftarrow [1, \boldsymbol{x}_i]$ for $\forall i$. ▷*Concatenate to incorporate the intercept*
$\boldsymbol{w}^+ \leftarrow \boldsymbol{w}^c$
$\mathcal{J} \leftarrow \{j : \lceil w_j^+ \rceil \neq \lfloor w_j^+ \rfloor\}$ ▷*Feature indices with fractional coefficients*
$\boldsymbol{\Gamma} \leftarrow [\lfloor \boldsymbol{w}^+ \rfloor; \lfloor \boldsymbol{w}^+ \rfloor; ...; \lfloor \boldsymbol{w}^+ \rfloor]^T$ ▷*n rows of* $\lfloor \boldsymbol{w}^+ \rfloor$
Define a new matrix $\boldsymbol{Z}$ with entries $Z_{ij} = y_i x_{ij}$
$\boldsymbol{\Gamma} \leftarrow \boldsymbol{\Gamma} + \mathbf{1}_{\boldsymbol{Z} \leq 0}$. ▷*See Theorem 3.1 and Second Inequality (Lipschitz continuity). This line performs the calculation:* $\gamma_{ij} = \lfloor w_j \rfloor$ *if* $y_i x_{ij} > 0$ *and* $\gamma_{ij} = \lceil w_j \rceil$ *otherwise.*
for i = 1 to n **do**
  $l_i \leftarrow 1/(1 + \exp(y_i \sum_{j=1}^p x_{ij} \Gamma_{ij}))$ ▷*Chosen so we can calculate local Lipschitz constant*
end for
while $\mathcal{J} \neq \emptyset$ **do** ▷*We iteratively round more coeffs in* $\boldsymbol{w}^+$ *until fractional coeffs are gone.*
  for $j \in \mathcal{J}$ **do** ▷*Try rounding both up and down for each* j
    $\boldsymbol{w}^{+j,up} \leftarrow (w_1^+, ..., \lceil w_j^+ \rceil, ...w_{p+1}^+)^T$, $\boldsymbol{w}^{+j,down} \leftarrow (w_1^+, ..., \lfloor w_j^+ \rfloor, ...w_{p+1}^+)^T$
    $U^{j,up} \leftarrow \sum_{i=1}^n (l_i \boldsymbol{x}_i^T (\boldsymbol{w}^{+j,up} - \boldsymbol{w}^c))^2$, $U^{j,down} \leftarrow \sum_{i=1}^n (l_i \boldsymbol{x}_i^T (\boldsymbol{w}^{+j,down} - \boldsymbol{w}^c))^2$
  end for
  ▷*Now find the best* j *and whether to round up or down.*
  $U^{up} \leftarrow \min_{j \in \mathcal{J}} U^{j,up}$, $U^{down} \leftarrow \min_{j \in \mathcal{J}} U^{j,down}$
  if $U^{up} \leq U^{down}$ **then**
    $j' \leftarrow \text{argmin}_{j \in \mathcal{J}} U^{j,up}$, $\mathcal{J} \leftarrow \mathcal{J} \setminus \{j'\}$
    $w_{j'}^+ \leftarrow \lceil w_{j'}^+ \rceil$ ▷*Round up*
  else
    $j' \leftarrow \text{argmin}_{j \in \mathcal{J}} U^{j,down}$, $\mathcal{J} \leftarrow \mathcal{J} \setminus \{j'\}$
    $w_{j'}^+ \leftarrow \lfloor w_{j'}^+ \rfloor$ ▷*Round down*
  end if
end while
$w_0^+ \leftarrow w^+[1], \boldsymbol{w}^+ \leftarrow \boldsymbol{w}^+[2 : end]$ ▷*Separate the intercept and the coefficients*
Return $(\boldsymbol{w}^+, w_0^+)$

B Comments on Proof of Chevaleyre *et al.*

Chevaleyre *et al.* [10] proposed Greedy Rounding, where coefficients are rounded sequentially. While this technique provides theoretical guarantees for greedy rounding for the hinge loss, we identified a serious flaw in their argument, rendering the bounds incorrect. We elaborate on this matter in this appendix.

The flaw is in the proof of Lemma 7. The proof essentially shows that for each sample i, there is at least one a (from the set $\{0, 1\}$) such that the inequality holds. However, the same a that works for sample $i = 3$ is not guaranteed to work for sample $i = 5$ for the inequality. It is not clear whether there exists one a that make all inequalities (for all samples i in $[1, ..., m]$) hold at the time.

To paraphrase, for each sample i, the proof shows that we can pick a set of a (either $\{0\}$, $\{1\}$, or $\{0, 1\}$) so that the inequality holds individually. However, we can not rule out the case that intersection of these individual sets is empty.

Without this extra argument, there is a gap between the statement of Lemma 7 and the proof of Lemma 7. Then, the bound for the greedy algorithm in Theorem 8 will not hold in the paper.

C Theoretical Upper Bound for the Rounding Method, Algorithm 6

The following theorem (as also shown in the main paper) states that we can provide an upper bound on the difference of the total loss between the integer solution $\boldsymbol{w}^+$ given by Algorithm 6 and the real-valued solution $\boldsymbol{w}$.

Theorem 3.1 (Loss incurred from rounding) Let $\boldsymbol{w}$ be the real-valued coefficients for the logistic regression model with objective function $L(\boldsymbol{w}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{x}_i^T \boldsymbol{w}))$. Let $\boldsymbol{w}^+$ be the integer-valued coefficients returned by the Auxiliaryloss Rounding method, Algorithm 6. Furthermore, let $u_j = w_j - \lfloor w_j \rfloor$. Let $l_i = 1/(1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{\gamma}_i))$ with $\gamma_{ij} = \lfloor w_j \rfloor$ if $y_i x_{ij} > 0$ and $\gamma_{ij} = \lceil w_j \rceil$ otherwise. Then, we have an upper bound on the difference between the loss $L(\boldsymbol{w})$ and the loss $L(\boldsymbol{w}^+)$:

$$L(\boldsymbol{w}^+) - L(\boldsymbol{w}) \leq \sqrt{n \sum_{i=1}^{n} \sum_{j=1}^{p} l_i^2 x_{ij}^2 u_j (1 - u_j)}. \tag{12}$$

To prove Theorem 3.1, we need to use the following Lemma C.1, which states that during each successive step of rounding a real-valued coefficient to the integer value, the deviation can be characterized and bounded by the data features and the real-valued coefficient.

Lemma C.1. *Suppose we have rounded the first $k-1$ real-valued coefficients to integers. Then for the k-th real-valued coefficient, if we set $w_k^+ = \operatorname{argmin}_{v \in \{\lfloor w_k \rfloor, \lceil w_k \rceil\}} \sum_{i=1}^{n} l_i^2 (\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) + x_{ik}(v - w_k))^2$, then we have*

$$\sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k} x_{ij}(w_j^+ - w_j) \right)^2 \leq \sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) \right)^2 + \sum_{i=1}^{n} l_i^2 x_{ik}^2 (1 - u_k) u_k \tag{13}$$

where $u_k = w_k - \lfloor w_k \rfloor$.

Proof. Let z_k be a binomial random variable so that $z_k = 1$ with probability u_j and $z_k = 0$ with probability $1 - u_k$. For notational convenience, let us define the function $f(v) := \sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) + x_{ik}(v - w_k) \right)^2$. Then $f(\lfloor w_k \rfloor + z_k)$ is a random variable, and the input to function $f(\cdot)$, which is $\lfloor w_k \rfloor + z_k$, takes on values either $\lfloor w_k \rfloor$ or $\lceil w_k \rceil$.

The expectation of this random variable is

$$\begin{aligned}
&\mathbb{E}_{z_k}\left[f(\lfloor w_k \rfloor + z_k)\right] \\
=&\mathbb{E}_{z_k}\left[\sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) + x_{ik}(\lfloor w_k \rfloor + z_k - w_k) \right)^2\right] \\
=&\sum_{i=1}^{n} l_i^2 \, \mathbb{E}_{z_k}\left[\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) + x_{ik}(\lfloor w_k \rfloor + z_k - w_k) \right)^2\right] && \textit{\# move } \mathbb{E}(\cdot) \textit{ inside the } \textstyle\sum(\cdot) \\
=&\sum_{i=1}^{n} l_i^2 \, \mathbb{E}_{z_k}\left[\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) + x_{ik}(z_k - u_k) \right)^2\right] && \textit{\# substitute with } u_k = w_k - \lfloor w_k \rfloor \\
=&\sum_{i=1}^{n} l_i^2 \left[\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) \right)^2 + 2x_{ik}\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j) \right) \mathbb{E}_{z_k}\left[z_k - u_k\right]\right. \\
&\left. + x_{ik}^2 \mathbb{E}_{z_k}\left[(z_k - u_k)^2\right]\right]. && \textit{\# expand the square term}
\end{aligned}$$

Notice that because $\mathbb{P}(z_k = 1) = u_k$, $\mathbb{P}(z_k = 0) = 1 - u_k$, we have

$$\mathbb{E}_{z_k}\left[z_k - u_k\right] = (1-u_k)u_k + (0-u_k)(1-u_k) = 0$$

and

$$\mathbb{E}_{z_k}\left[(z_k - u_k)^2\right] = (1-u_k)^2 u_k + (0-u_k)^2(1-u_k) = u_k(1-u_k). \qquad \textit{\# similar as above}$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{z_k}\left[f(\lfloor w_k \rfloor + z_k)\right] \\
=&\sum_{i=1}^{n} l_i^2 \left[\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 \right. \\
&\qquad \left. +2x_{ik}\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)\mathbb{E}_{z_k}\left[z_k - u_k\right] + x_{ik}^2 \mathbb{E}_{z_k}\left[(z_k-u_k)^2\right]\right] \\
=&\sum_{i=1}^{n} l_i^2 \left[\left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 + x_{ik}^2 u_k(1-u_k)\right] \qquad \textit{\# plug in the two expectations above} \\
=&\sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 + \sum_{i=1}^{n} l_i^2\, x_{ik}^2 u_k(1-u_k). \qquad \textit{\# split into two summation terms}
\end{aligned}$$

Since the expectation of $f(\lfloor w_k \rfloor + z_k)$ is equal to $\sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 + \sum_{i=1}^{n} l_i^2\, x_{ik}^2 u_k(1-u_k)$, there exists a $z_k' \in \{0,1\}$ such that

$$f(\lfloor w_k \rfloor + z_k') \le \sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 + \sum_{i=1}^{n} l_i^2\, x_{ik}^2 u_k(1-u_k). \tag{14}$$

Note that $\lfloor w_k \rfloor + z_k'$ is the minimizer of $f(\cdot)$ because the other input value $\lfloor w_k \rfloor + 1 - z_k'$ will take the value $f(\lfloor w_k \rfloor + 1 - z_k')$, which is greater than or equal to the expectation $\mathbb{E}_{z_k}[f(\lfloor w_k \rfloor + z_k)]$.

If we round w_k to an integer by setting $w_k^+ = \lfloor w_k \rfloor + z_k'$, then $w_k^+ = \operatorname{argmin}_{v \in \{\lfloor w_k \rfloor, \lceil w_k \rceil\}} f(v)$. We now have:

$$\begin{aligned}
\sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k} x_{ij}(w_j^+ - w_j)\right)^2 &= \min_{v \in \{\lfloor w_k \rfloor, \lceil w_k \rceil\}} f(v) && \textit{\# definition of } w_k^+ \text{ and } f(\cdot) \\
&= \min_{c \in \{0,1\}} f(\lfloor w_k \rfloor + c) && \textit{\# substitute } v = \lfloor w_k \rfloor + c \\
&= f(\lfloor w_k \rfloor + z_k') && \textit{\# } \lfloor w_k \rfloor + z_k' \textit{ is the minimizer of } f(\cdot) \\
&\le \sum_{i=1}^{n} l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 + \sum_{i=1}^{n} l_i^2 x_{ik}^2 (1-u_k)u_k, \\
& && \textit{\# Inequality 14}
\end{aligned}$$

thus completing our proof.

□

Now we can use Lemma C.1 to prove Theorem 3.1.

Proof of Theorem 3.1. For simplicity, let us first consider the case where we round coefficients sequentially from w_1^+ to w_p^+. We claim that if at each step r, we round $w_r^+ =$

$\text{argmin}_{v\in\{\lfloor w_r\rfloor,\lceil w_r\rceil\}} \sum_{i=1}^n l_i^2 \left(\sum_{j=1}^{l-1} x_{ij}(w_j^+ - w_j) + x_{ir}(v - w_r)\right)^2$, then for $\forall k \in [1, ..., p]$

$$\sum_{i=1}^n l_i^2 \left(\sum_{j=1}^k x_{ij}(w_j^+ - w_j)\right)^2 \leq \sum_{i=1}^n \sum_{j=1}^k l_i^2 x_{ij}^2 u_j(1-u_j). \tag{15}$$

We prove this by the principle of induction. Suppose for step $k-1$, we have

$$\sum_{i=1}^n l_i^2 \left(\sum_{j=1}^{k-1} x_{ij}(w_j^+ - w_j)\right)^2 \leq \sum_{i=1}^n \sum_{j=1}^{k-1} l_i^2 x_{ij}^2 u_j(1-u_j).$$

Then, according to Lemma C.1 and the previous line, we have

$$\begin{aligned}
\sum_{i=1}^n l_i^2 \left(\sum_{j=1}^k x_{ij}(w_j^+ - w_j)\right)^2 &\leq \sum_{i=1}^n \sum_{j=1}^{k-1} l_i^2 x_{ij}^2 u_j(1-u_j) + \sum_{i=1}^n l_i^2 x_{ik}^2 u_k(1-u_k) && \textit{\# Lemma C.1} \\
&= \sum_{i=1}^n \sum_{j=1}^k l_i^2 x_{ij}^2 u_j(1-u_j). && \textit{\# use a single sum } \textstyle\sum_{i=1}^n(\cdot)
\end{aligned}$$

For the base step $k = 1$, Lemma C.1 also implies that

$$\sum_{i=1}^n l_i^2 (x_{i1}(w_1^+ - w_1))^2 \leq \sum_{i=1}^n l_i^2 x_{i1}^2 u_1(1-u_1).$$

Thus, Inequality (15) works for all k. If we let $k = p$, we have

$$\sum_{i=1}^n l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j)\right)^2 \leq \sum_{i=1}^n \sum_{j=1}^p l_i^2 x_{ij}^2 u_j(1-u_j). \tag{16}$$

Also, notice that this inequality holds for sequential rounding of any permutation of the feature indices $[1, ..., p]$, and the rounding order of the AuxiliaryLossRounding method is one specific feature order. Therefore, the Inequality (16) works for the AuxiliaryLossRounding method as well.

Lastly, we use Inequality 16 to derive an upper bound on the logistic loss of the AuxiliaryLossRounding method. Recall that our objective is:

$$L(\boldsymbol{w}) = \sum_{i=1}^n \log(1 + \exp(-y_i \boldsymbol{x}_i^T \boldsymbol{w})). \tag{17}$$

The loss difference between the rounded solution and the real-valued solution can be bounded as follows:

$$\begin{aligned}
L(\boldsymbol{w}^+) - L(\boldsymbol{w}^*) &\leq \sum_{i=1}^n \left[\log(1 + \exp(-y_i \boldsymbol{x}_i^T \boldsymbol{w}^+)) - \log(1 + \exp(-y_i \boldsymbol{x}_i^T \boldsymbol{w}))\right] \\
&\leq \sum_{i=1}^n |l_i(y_i \boldsymbol{x}_i^T \boldsymbol{w}^+ - y_i \boldsymbol{x}_i^T \boldsymbol{w})| && \textit{\# Lipschitz continuity, see details below} \\
&= \sum_{i=1}^n |l_i y_i \boldsymbol{x}_i^T (\boldsymbol{w}^+ - \boldsymbol{w})| && \textit{\# pull out common factor} \\
&= \sum_{i=1}^n |l_i \boldsymbol{x}_i^T (\boldsymbol{w}^+ - \boldsymbol{w})| && \textit{\# since } |y_i| = 1 \\
&\leq \sum_{i=1}^n \sqrt{l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j)\right)^2} && \textit{\# rewrite } |\cdot| \textit{ in terms of } \sqrt{\cdot} \\
&\leq \sqrt{n \sum_{i=1}^n l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j)\right)^2} && \textit{\# Jensen's Inequality, see details below}
\end{aligned}$$

There are two inequalities we need to elaborate in details, the second and the last inequalities (Lipschitz continuity and Jensen's Inequaltiy).

Second Inequality (Lipschitz continuity):

The second inequality holds because the logistic loss $g(a) = \log(1 + \exp(-a))$ is Lipschitz continuous. If the Lipschitz constant is l, then we have $|g(a) - g(b)| \leq l\ |a - b|$. We now explain how we derive the Lipschitz constant $l_i = 1/(1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{\gamma}_i))$ with $\gamma_{ij} = \lfloor w_j \rfloor$ if $y_i x_{ij} > 0$ and $\gamma_{ij} = \lceil w_j \rceil$ as stated in Theorem 3.1.

Since the logistic loss function $g(\cdot)$ is differentiable, the smallest Lipschitz constant of the function $g(\cdot)$ is $l_{\min}(g) = \sup_{a \in \text{Domain}(g)} |g'(a)|$. To see this, by the definition of the Lipschitz constant, we have $\frac{|g(a)-g(b)|}{|a-b|} \leq l$. If we take the limit $b \to a$, the inequality still holds, $\lim_{b \to a} \frac{|g(a)-g(b)|}{|a-b|} \leq l$. The left hand side converges to the absolute value of the derivative of $g(\cdot)$ at a. Therefore, we have $|g'(a)| \leq l$. Since this works for all a, and we want to find the smallest Lipschitz value, we have $l_{\min}(g) = \sup_{a \in \text{Domain}(g)} |g'(a)|$.

For the logistic loss $g(a) = \log(1 + e^{-a})$, the absolute value of the derivative is $|g'(a)| = \frac{1}{1+e^a}$. Thus, if a is lower-bounded so that $a \geq a_1$, the smallest Lipschitz constant of the logistic loss is $l_{\min}(g) = \frac{1}{1+e^{a_1}}$.

We can apply this fact to calculate a smaller Lipschitz constant for each sample's term. If $\gamma_{ij} := \lfloor w_j \rfloor$ if $y_i x_{ij} > 0$ and $\gamma_{ij} := \lceil w_j \rceil$ otherwise, then

$$y_i \boldsymbol{x}_i \boldsymbol{w}^+ \geq y_i \boldsymbol{x}_i^T \boldsymbol{\gamma}_i, \text{ and } |g'(y_i \boldsymbol{x}_i \boldsymbol{w}^+)| \leq 1/(1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{\gamma}_i)).$$

Therefore, $l_i = 1/(1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{\gamma}_i))$ is a valid Lipschitz constant for the i-th sample.

Last Inequality (Jensen's inequality):

Jensen's Inequality states that $\mathbb{E}_z[\phi(g(z))] \geq \phi(\mathbb{E}_z[g(z)])$ for any convex function $\phi(\cdot)$. For this specific problem, let $\phi(b) = -\sqrt{b}$ and let $g(z) = l_i^2 (\sum_{j=1}^p x_{ij}(w_j^+ - w_j))^2$ for a particular i with probability $\frac{1}{n}$. Then, we have

$$\begin{aligned}
\sum_{i=1}^n \sqrt{l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j) \right)^2} &= n \sum_{i=1}^n \frac{1}{n} \sqrt{l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j) \right)^2} \\
& \qquad \textit{\# multiply and divide by } n \\
&= -n \mathbb{E}_z[\phi(g(z))] \qquad \textit{\# definition of } \phi(\cdot),\ g(\cdot), \text{ and } \mathbb{E}(\cdot) \\
&\leq -n \phi(\mathbb{E}_z[g(z)]) \qquad \textit{\# Jensen's Inequality} \\
&= n \sqrt{\frac{1}{n} \sum_{i=1}^n l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j) \right)^2} \\
& \qquad \textit{\# write out } \phi(\cdot),\ g(\cdot), \text{ and } \mathbb{E}(\cdot) \text{ explicitly} \\
&= \sqrt{n \sum_{i=1}^n l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j) \right)^2}. \qquad \textit{\# move } n \textit{ inside } \sqrt{\cdot}
\end{aligned}$$

Therefore, using Inequality 16, we can now bound the loss difference between the rounded solution and the real-valued solution as stated in Theorem 3.1:

$$\begin{aligned}
L(\boldsymbol{w}^+) - L(\boldsymbol{w}) &\leq \sqrt{n \sum_{i=1}^n l_i^2 \left(\sum_{j=1}^p x_{ij}(w_j^+ - w_j) \right)^2} \\
&\leq \sqrt{n \sum_{i=1}^n \sum_{j=1}^p l_i^2 x_{ij}^2 u_j (1 - u_j)}.
\end{aligned}$$

□

D Experimental Setup

D.1 Dataset Information

The dataset names, data source, number of samples and features, and the classification tasks can be found in Table 2. The datasets with results shown in the main paper (adult, bank, breast-cancer, mammo, mushroom, spambase) were directly downloaded from this link: https://github.com/ustunb/risk-slim/tree/master/examples/data. The COMPAS dataset can be downloaded from this link: https://github.com/propublica/compas-analysis/blob/master/compas-scores-two-years.csv. The FICO dataset can be requested and downloaded from this website: https://community.fico.com/s/explainable-machine-learning-challenge. The Netherlands dataset is available through Data Archiving and Networked Services https://easy.dans.knaw.nl/ui/datasets/id/easy-dataset:78692.

For our experiments on the COMPAS, FICO, and Netherlands datasets, we convert the continuous features into a set of highly correlated dummy variables, with all entries equal to 1 or 0. By conducting experiments on these three datasets, we can test how well FasterRisk works for highly correlated features. We use the preprocessing steps as explained in Section C2 of [24]. We list the key preprocessing steps below.

COMPAS: In addition to the label *"two_year_recid"*, we use features *"sex", "age", "juv_fel_count", "juv_misd_count", "juv_other_count", "priors_count"*, and *"c_charge_degree"*.

FICO: All continuous features are used.

Netherlands: In addition to the label *"recidivism_in_4y"*, we use features *"sex", "country of birth", "log # of previous penal cases", "11-20 previous case", and ">20 previous case", "age in years", "age at first penal case"*, and *"offence type"*.

For each continuous variable $x_{\cdot,j}$, it is converted into a set of highly correlated dummy variables $\tilde{x}_{\cdot,j,\theta} = \mathbf{1}_{[x_{\cdot,j} \leq \theta]}$, where θ are all unique values that have appeared in feature column j. For Netherlands, special preprocessing steps are performed for *"age in years"* (which is real-valued, not integer) and *"age at first penal case"*. Instead of considering all unique values in the feature column, we consider 1000 quantiles.

Dataset	Source	N	P	Classification task
adult	[20]	32561	36	Predict if a U.S. resident earns more than $50,000
bank	[28]	41188	55	Predict if a person opens account after marketing call
breastcancer	[26]	683	9	Detect breast cancer using a biopsy
mammo	[15]	961	14	Detect breast cancer using a mammogram
mushroom	[29]	8124	113	Predict if a mushroom is poisonous
spambase	[12]	4601	57	Predict if an e-mail is spam
COMPAS	[21]	6907	134	Predict if someone will be arrested $\leq$ 2 years of release
FICO	[17]	10459	1917	Predict if someone will default on a loan
Netherlands	[36]	20000	2024	Predict if someone will have any charge within 4 years

Table 2: Dataset information. Breastcancer and spambase datasets have real-valued features. All other datasets have binary (0 or 1) features.

D.2 Computing Platform

We ran all experiments on a TensorEX TS2-673917-DPN Intel Xeon Gold 6226 Processor with 2.7Ghz (768GB RAM 48 cores). For all experiments, we used only two cores because we observed using more cores did not improve the computational speed further.

D.3 Baselines

We compare with several baselines in our experiments.

RiskSLIM The current state-of-the-art method is RiskSLIM. We installed this package from the following GitHub link: `https://github.com/ustunb/risk-slim`[2]. RiskSLIM uses the IBM CPLEX MIP solver to do the optimization. The CPLEX version we used is 12.8.

Pooled Approaches For other baselines, we first found a pool of continuous sparse solutions by the ElasticNet [48] method and then rounded the coefficients to integers with different rounding techniques. Because ElasticNet has ℓ_1 and ℓ_2 penalties, we call this method the penalized logistic regression (PLR) approach. The best integer solution was selected from this pool based on which solution produces the smallest logistic loss while obeying the sparsity constraint and box constraints. These baselines correspond to the Pooled Approaches in Section 5.1 of [39], where Figure 11 and Figure 12 clearly show that pooled approaches are much better than traditional approaches. We include Unit Weighting and Rescaled Rounding as two additional rounding methods. The details of the pooled approach and the rounding techniques can be found in Section 5.1 of [39].

The ElasticNet method tries to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{2n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{x}_i^T \boldsymbol{w})) + \lambda \cdot (\alpha \|\boldsymbol{w}\|_1 + (1-\alpha)\|\boldsymbol{w}\|_2^2) \tag{18}$$

where $\alpha \in$ [0,1] is a hyperparameter. By controlling α, we choose the best model over Ridge (α =0), Lasso (α=1), and Elastic net ($0 < \alpha < 1$). We generated 1,100 models using the glmnet package[3]. To do this, we first choose 11 values of $\alpha \in \{0, 0.1, 0.2, ..., 0.9, 1.0\}$. For each given α, the package then internally and automatically selects 100 λ's equi-spaced on the logarithmic scale between $\lambda_{\min}$ and $\lambda_{\max}$ (the smallest value for λ such that all the coefficients are zero). We call this part the *Pooled-PLR* (Pooled Penalized Logistic Regression).

To convert each continuous sparse model to an integer sparse model, we applied the following rounding methods:

- 1) Pooled-PLR-RD: For each of the 1,100 PLR models in the pool, we first truncated all the coefficients (except the intercept β_0) to be within the range [-5,5] and did simple rounding: $\beta_j = \lceil \min(\max(\beta_j, -5), 5) \rfloor$, and $\beta_0 = \lceil \beta_0 \rfloor$. The $\lceil \cdot \rfloor$ operation is defined as $\lceil a \rfloor = \lceil a \rceil$ if $|a - \lceil a \rceil| < |a - \lfloor a \rfloor|$ and $\lceil a \rfloor = \lfloor a \rfloor$ otherwise.
- 2) Pooled-PLR-RDU: For each solution, we rounded each of its coefficients to be ± 1 based on its signs: $w_j = \text{sign}(w_j)\mathbb{1}_{[w_j \neq 0]}$ and $w_0 = \lceil w_0 \rfloor$ This rounding technique is known as unit weighting or the Burgess method.
- 3) Pooled-PLR-RSRD: For each solution, we rescaled its coefficients by a factor γ so that $\gamma w_{\max} = \pm 5$ and then rounded each rescaled coefficient to the nearest integer: $w_j = \lceil \gamma w_j \rfloor, \gamma = \frac{5}{\max_j |\lambda_j|}$ and $w_0 = \lceil w_0 \rfloor$.
- 4) Pooled-PLR-Rand: For each model in the pool, for each coefficient, denote its fractional part by $u_j = w_j - \lfloor w_j \rfloor$. We rounded each coefficient up to $\lceil w_j \rceil$ with probability u_j and down to $\lceil w_j \rceil$ with probability $1 - u_j$. After all rounding was done, we selected the best model in the pool.
- 5) Pooled-PLR-RDP: For each model in the pool, we iterated through each coefficient β_j and calculated the loss for both $\lceil \beta_j \rceil$ and $\lfloor \beta_j \rfloor$ and selected the rounding that minimizes the loss. This is called Sequential Rounding in [39].
- 6) Pooled-PLR-RDSP: we first rounded through Sequential Rounding (Method 5, just above), and then we applied Discrete Coordinate Descent (DCD) [39] to iteratively improve the loss by adjusting one coefficient at a time. At each round, DCD selects the coefficient and its new value that decreases the logistic loss the most.

As mentioned earlier, after we get the 1,100 integer sparse models via each rounding technique, we selected the best model from the pool based on which solution has the smallest logistic loss.

[2]The license for this package is BSD 3-Clause license. The license can be viewed on the GitHub page.

[3]We installed the package from the following GitHub link: `https://github.com/bbalasub1/glmnet_python` The package contains GNU license, which can be viewed on the GitHub website.

D.4 Hyperparameters Specification

We used the default values in Algorithm 1 for all hyperparameters. We reiterate the hyperparameters used in the experiments below.

- beam search size: $B = 10$.
- tolerance level for sparse diverse pool: $\epsilon = 0.3$ (or 30%).
- number of attempts to try for sparse diverse pool: $T = 50$.
- number of multipliers to try: $N_m = 20$.

Performance is not particularly sensitive to these choices (see Appendix E.10). If T, N_m, B are chosen too large, the algorithm will take longer to execute.

E Additional Experimental Results

E.1 Additional Results on Solution Quality

In addition to the six datasets we show in the main paper, we provide results on the breastcancer, spambase, and Netherlands datasets (see Section D.1 for more data information). The comparison of solution quality is shown in Figure 6. We see that FasterRisk outperforms both RiskSLIM and other pooled approaches, even with high dimensional feature spaces and in the presence of highly correlated features (the Netherlands dataset).

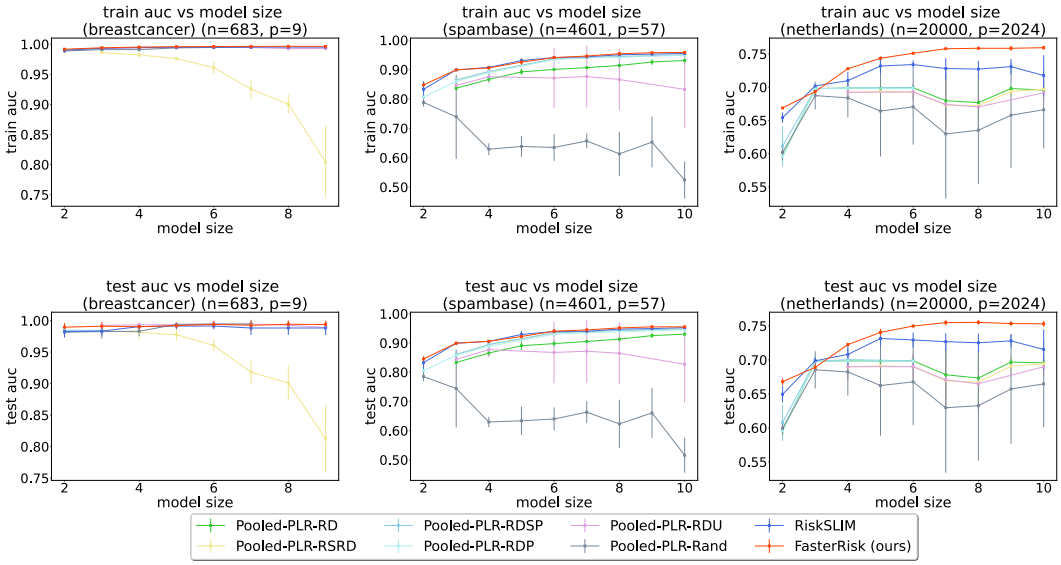

Figure 6: Performance comparison on the breastcancer, spambase, and Netherlands datasets. Top row is training AUC (higher is better) and bottom row is test AUC (higher is better).

E.2 Additional Results on Direct Comparison with RiskSLIM

As RiskSLIM provides state-of-the-art performance, we compare it to FasterRisk in isolation to highlight the differences between the two approaches/algorithms. The results are shown in Figure 7 on the breastcancer, spambase, and Netherlands datasets.

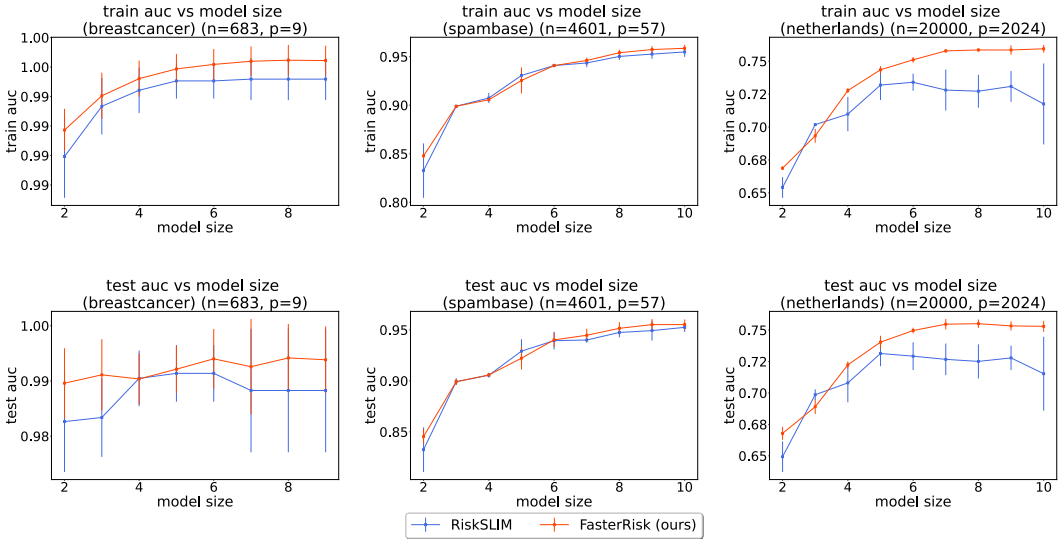

Figure 7: Detailed performance comparison between FasterRisk and RiskSLIM on breastcancer, spambase, and Netherlands. Top row is training AUC (higher is better) and bottom row is test AUC (higher is better). We can improve FasterRisk's results on the spambase dataset by increasing the beam size in the algorithm. See Figure 29 for the perturbation study on this hyperparameter.

E.3 Additional Results on Running Time

We also provide a runtime comparison between RiskSLIM and FasterRisk in Figure 8. Except for the small dataset breastcancer, RiskSLIM timed out in all other instances. In contrast, FasterRisk finishes running under 50s or 100s on all cases, showing great scalability, even in high dimensional feature space and in presence of highly correlated features (the Netherlands dataset).

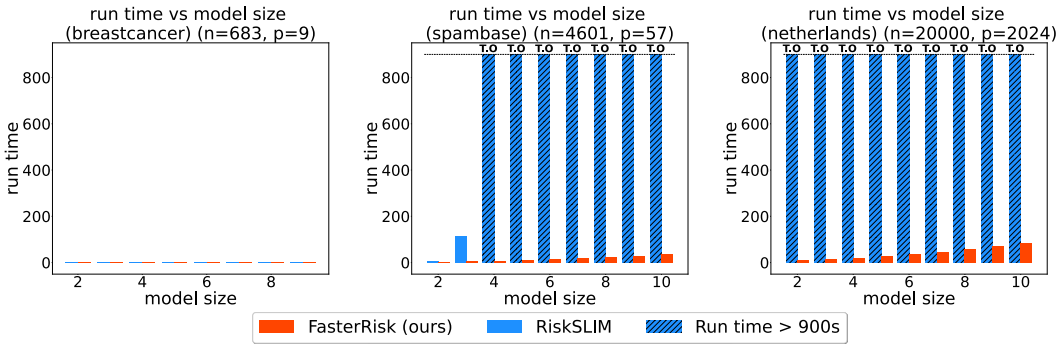

Figure 8: Runtime Comparison. Runtime (in seconds) versus model size for our method FasterRisk (in red) and the RiskSLIM (in blue). The shaded blue bars indicate cases that timed out ("T.O") at 900 seconds.

E.4 Ablation Study of the Proposed Techniques

We investigate how each component of FasterRisk, including sparse beam search, diverse pool, and multipliers, contribute to solution quality. We quantify the contribution of each part of the algorithm by means of an ablation study in which we run variations of FasterRisk, each with a single component disabled.

The results are shown in Figure 9-11. "no beam search" means that the beam size is 1, so we expand the support by picking the next feature based on which new feature can induce the smallest logistic loss via the single coordinate optimization. "no sparse diverse" means that the sparse diverse pool contains only the solution by Algorithm 2, the SparseBeamLR method. "no multiplier" means that there is no "star ray search" of the multiplier. There is no scaling of coefficients or the data, so we think of this as setting multiplier to 1.

The ablation study shows that different parts of our algorithm provide the biggest benefit to different data sets — that is, there is no single component of the algorithm that uniformly assists with performance; instead, the combination of these techniques, working in concert, is responsible. We provide the detailed analysis of the contributions for each specific dataset in the figure captions.

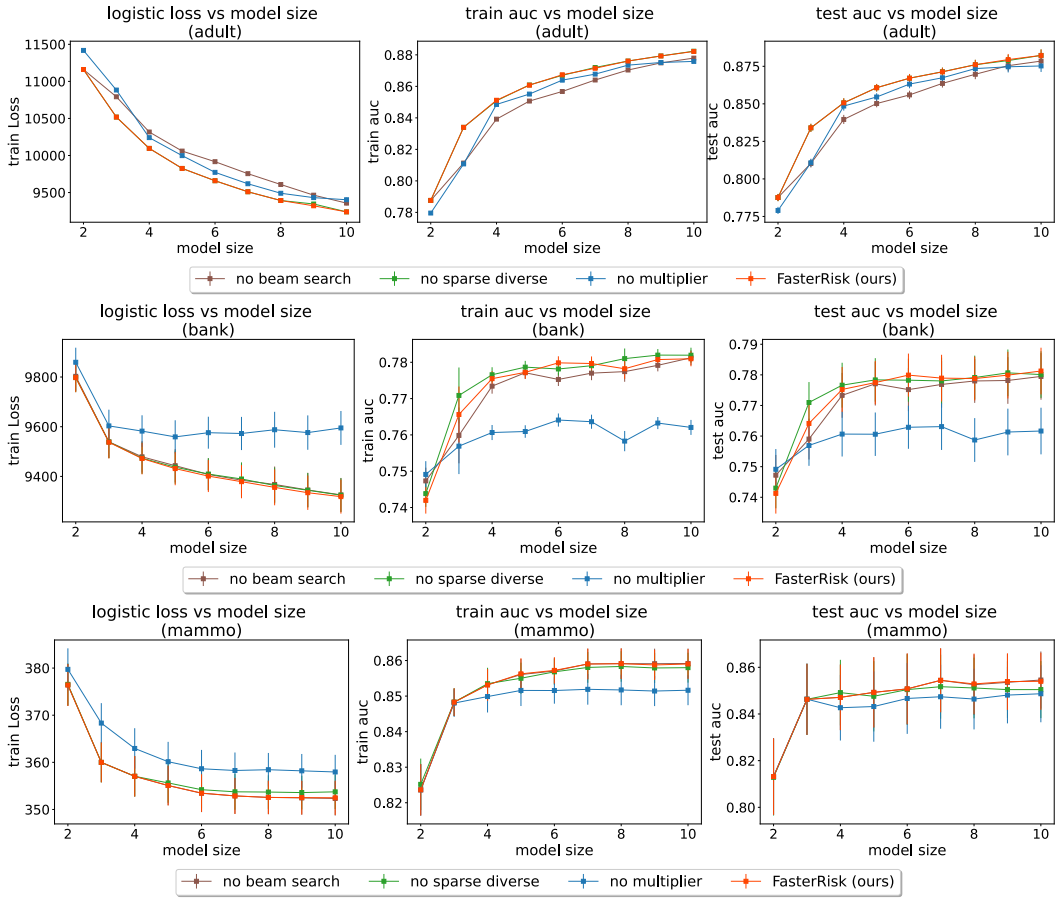

Figure 9: Ablation study on the adult, bank, and mammo datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). The "beam search" method is particularly helpful on the adult dataset. The use of "multiplier" is particularly helpful on all three datasets. The "diverse pool" technique is somewhat helpful on the mammo dataset. More significant contributions from "diverse pool" can be found in Figure 11 and Figure 10.

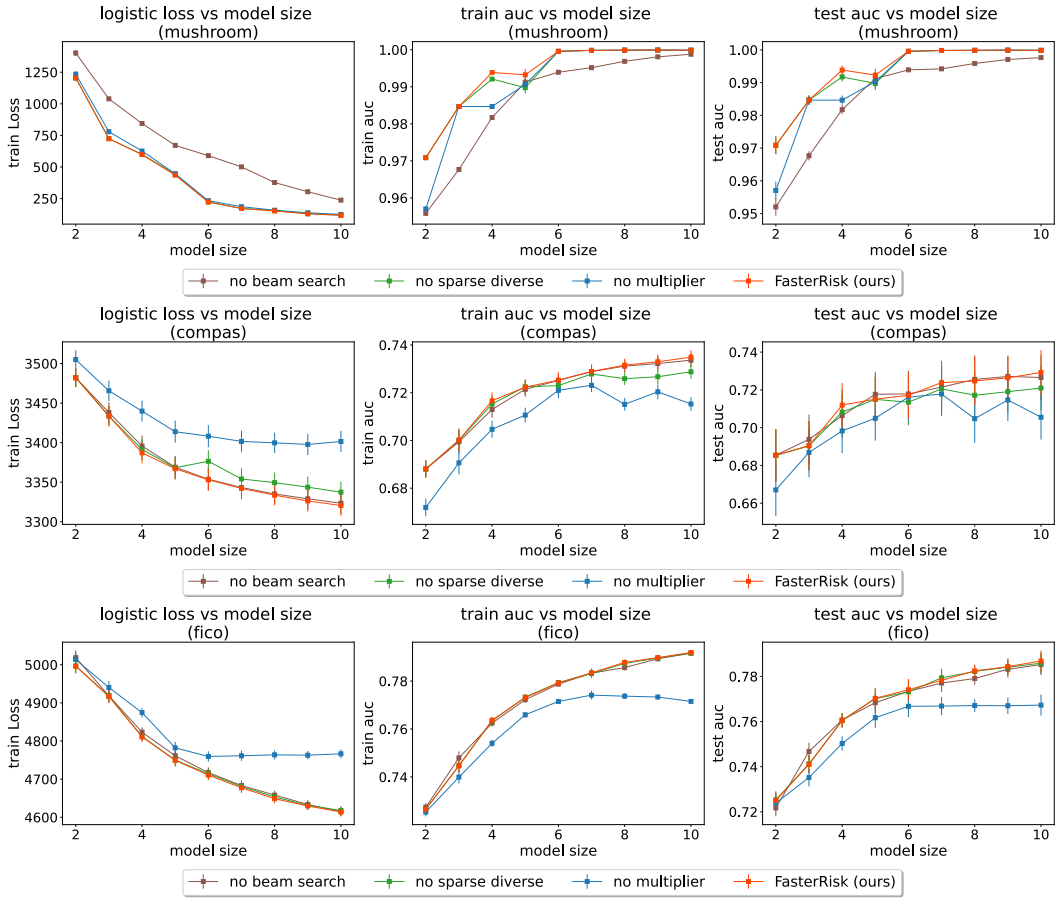

Figure 10: Ablation study on the mushroom, COMPAS, and FICO datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). The "beam search" method is particularly helpful on the mushroom dataset. The use of "multiplier" is particularly helpful on the COMPAS and FICO datasets. The "diverse pool" technique is particularly helpful on the COMPAS dataset.

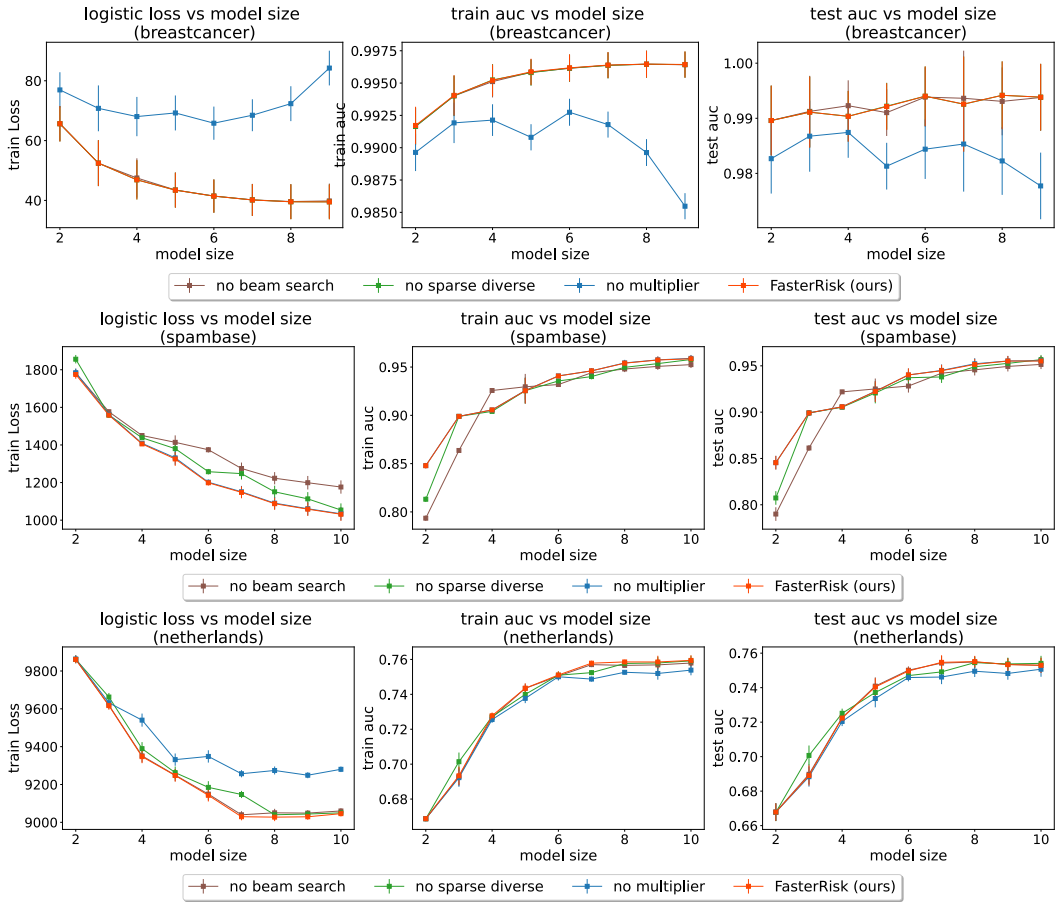

Figure 11: Ablation study on the COMPAS, FICO, and Netherlands datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). The "beam search" method is particularly helpful on the spambase dataset. The use of "multiplier" is particularly helpful on breastcancer and netherlands datasets. The "diverse pool" technique is particularly helpful on the spambase and Netherlands datasets.

E.5 Training Losses of FasterRisk vs. RiskSLIM

In the main paper, due to the page limit, we have only compared the training and test AUCs between RiskSLIM and our FasterRisk. Here, we provide the comparison of training loss (logistic loss) between these two methods. The results are shown in Figure 12. We can see that FasterRisk outperforms RiskSLIM in almost all model size instances and datasets.

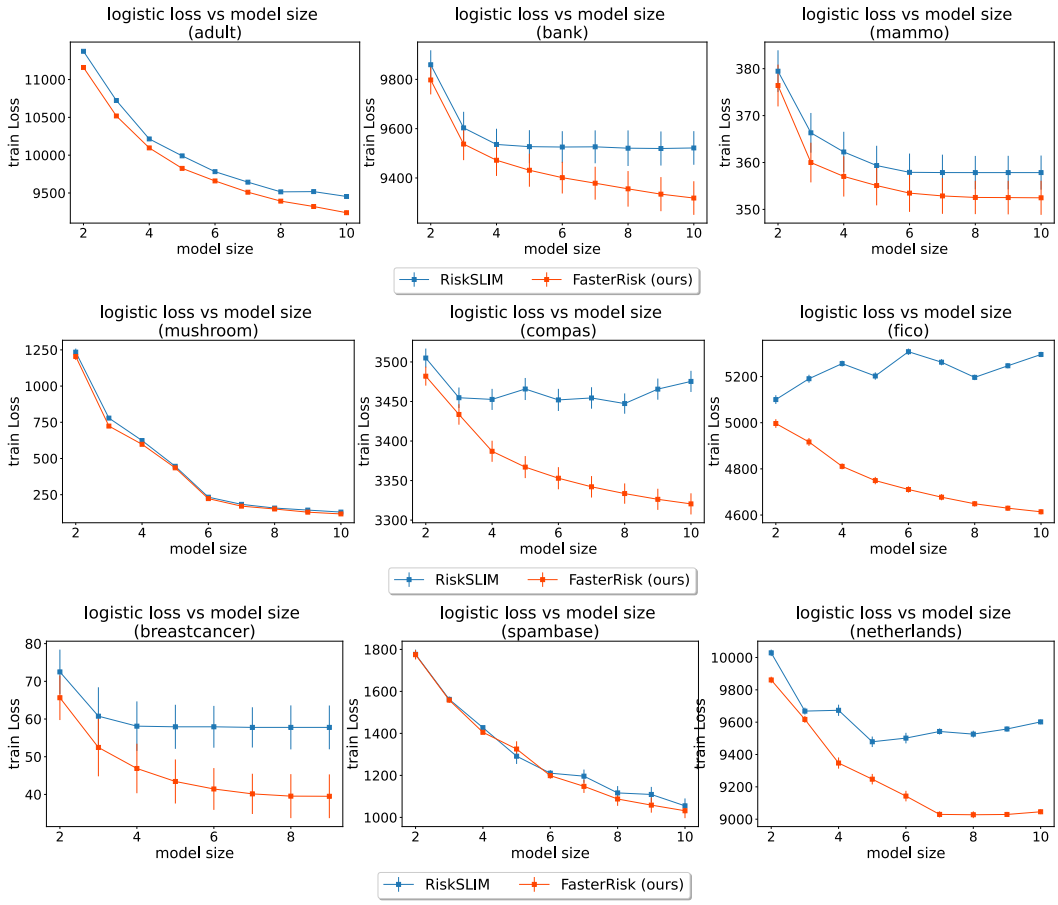

Figure 12: Training loss between RiskSLIM and our FasterRisk methods. (lower is better)

E.6 Comparison of SparseBeamLR with OMP and fastSparse

We next study how effective SparseBeamLR is in producing continuous sparse coefficients under the ℓ_0 sparsity and box constraints. We compare with two existing methods, OMP [14] and fastSparse [24]. OMP stands for Orthogonal Matching Pursuit, which expands the support by selecting the next feature with the largest magnitude of partial derivative. fastSparse tries to solve the logistic loss objective with an ℓ_0 regularization. For fastSparse, we use the default λ_0 values (coefficient for the ℓ_0 regularization) internally selected by the software. Specifically, the software first apply a large λ_0 value to produce a super-sparse solution (with support size equal to 1 or close to 1). Then, in the solution path, the λ_0 value is sequentially decreased until the produced sparse model violates the model size constraint.

The results are shown in Figure 13-15. Although OMP and fastSparse can somtimes produce high-quality solutions on some model size instances and datasets, SparseBeamLR is the only method that consistently produces high-quality sparse solutions in all cases.

OMP's solution quality is usually worse than that of SparseBeamLR, and OMP could not produce coefficients that satisfy the box constraints on the mushroom and spambase datasets.

fastSparse also cannot produce coefficients that satisfy the box constraints on the mushroom and spambase datasets. Additionally, it is hard to control the λ_0 regularization to produce the exact model size desired. In Figure 14, we do not obtain any model with model size equal to 9 or 10 in the solution path.

The limitations of OMP and fastSparse stated above are our main motivations for developing the SparseBeamLR method.

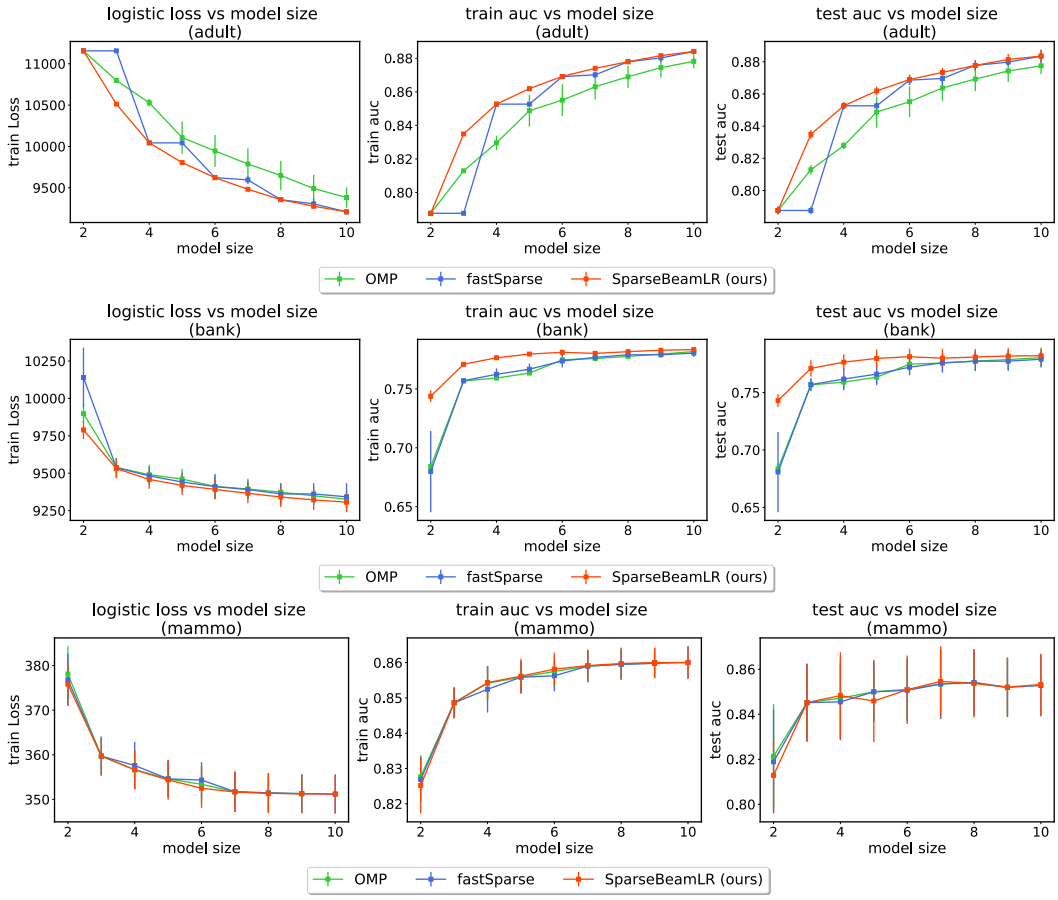

Figure 13: Sparse continuous solutions on the adult, bank, and mammo datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). SparseBeamLR consistently produces high-quality continuous sparse solutions.

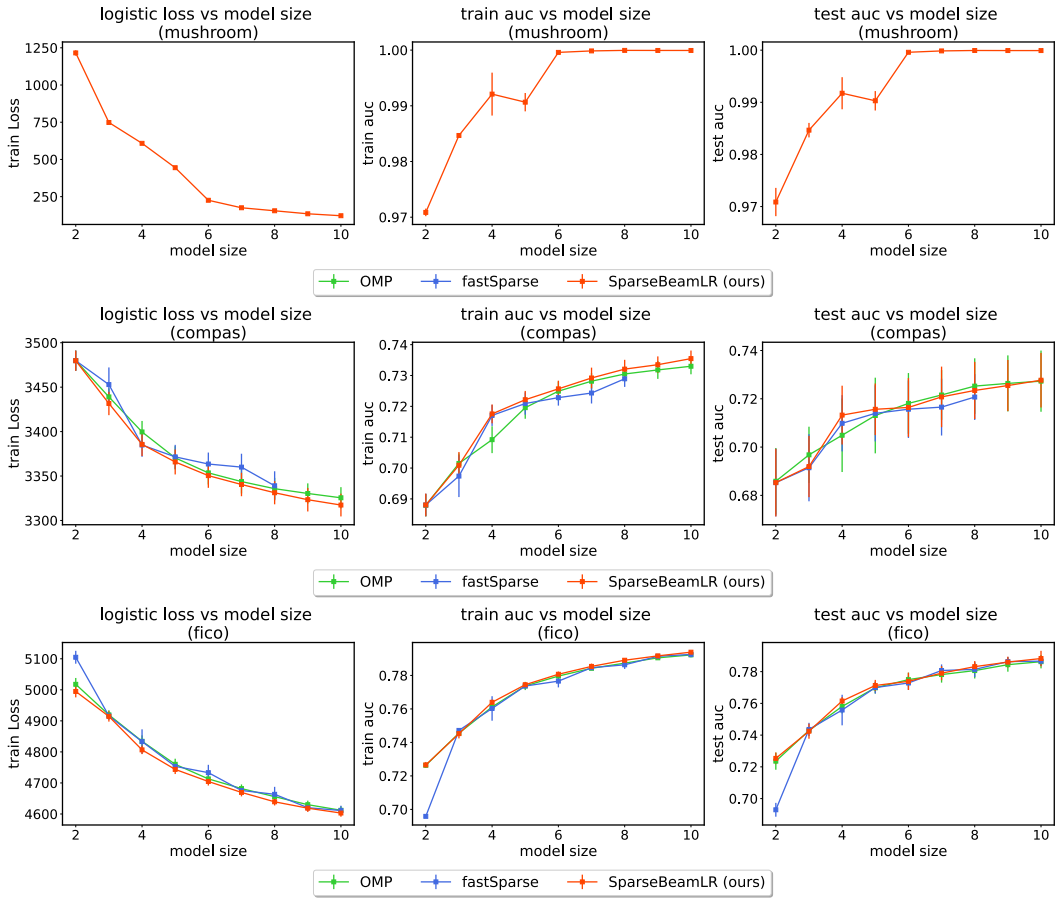

Figure 14: Sparse continuous solutions on the mushroom, COMPAS, and FICO datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). The solution coefficients by the OMP and fastSparse methods violate the box constraints on the mushroom dataset, so we omit the results in the plot. fastSparse cannot obtain solutions with model size equal to 9 or 10 on the COMPAS dataset, so we do not show those points in the plot.

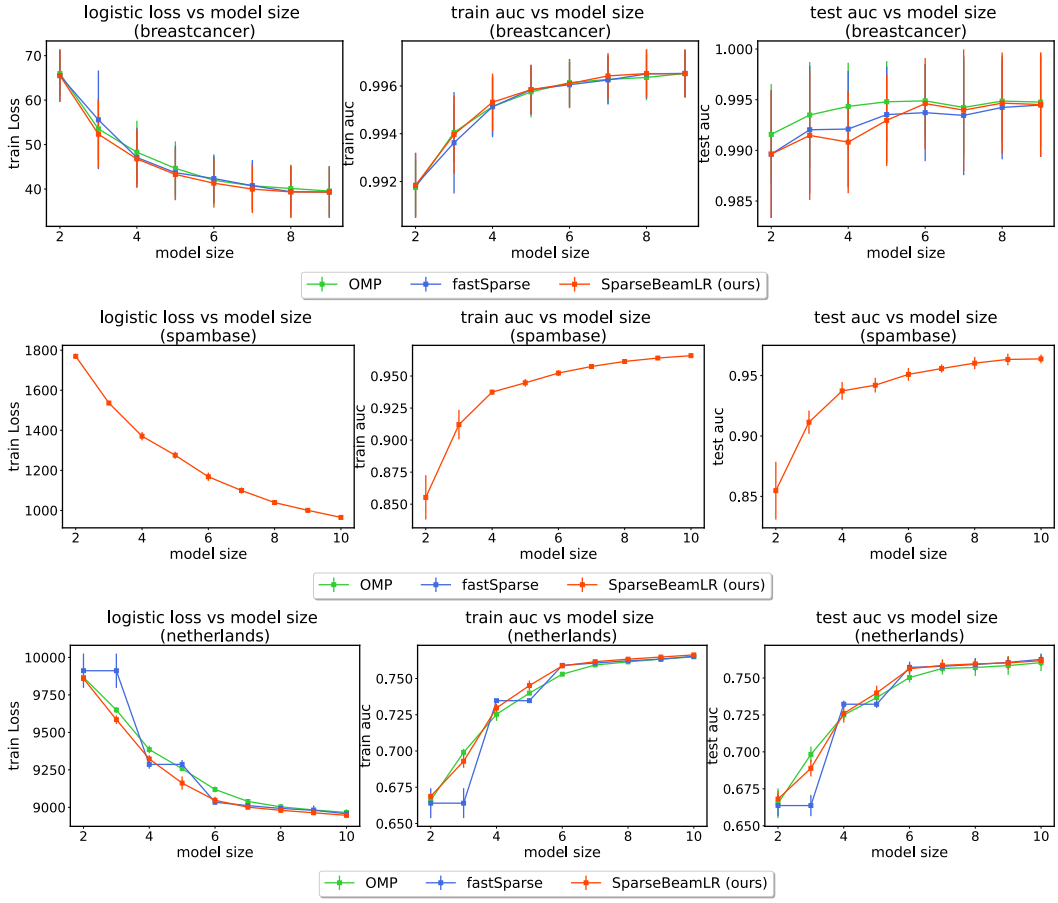

Figure 15: Sparse continuous solutions on the breastcancer, spambase, and Netherlands datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). SparseBeamLR consistently produces high-quality continuous sparse solutions. The solution coefficients of OMP and fastSparse violate the box constraints on the spambase dataset, so we omit the results in the plot.

E.7 Comparison of FasterRisk with OMP (or fastSparse) + Sequential Rounding

Having compared the continuous sparse solutions, we next compare the integer sparse solutions produced by OMP, fastSparse, and FasterRisk. After obtaining the continuous sparse solutions from OMP and fastSparse from Section E.6, we round the continuous coefficients to integers using the Sequential Rounding method as stated in Method 5 of D.3.

The results are shown in Figure 16-18. FasterRisk consistently outperforms the other two methods, due to higher quality of continuous sparse solutions and the use of multipliers.

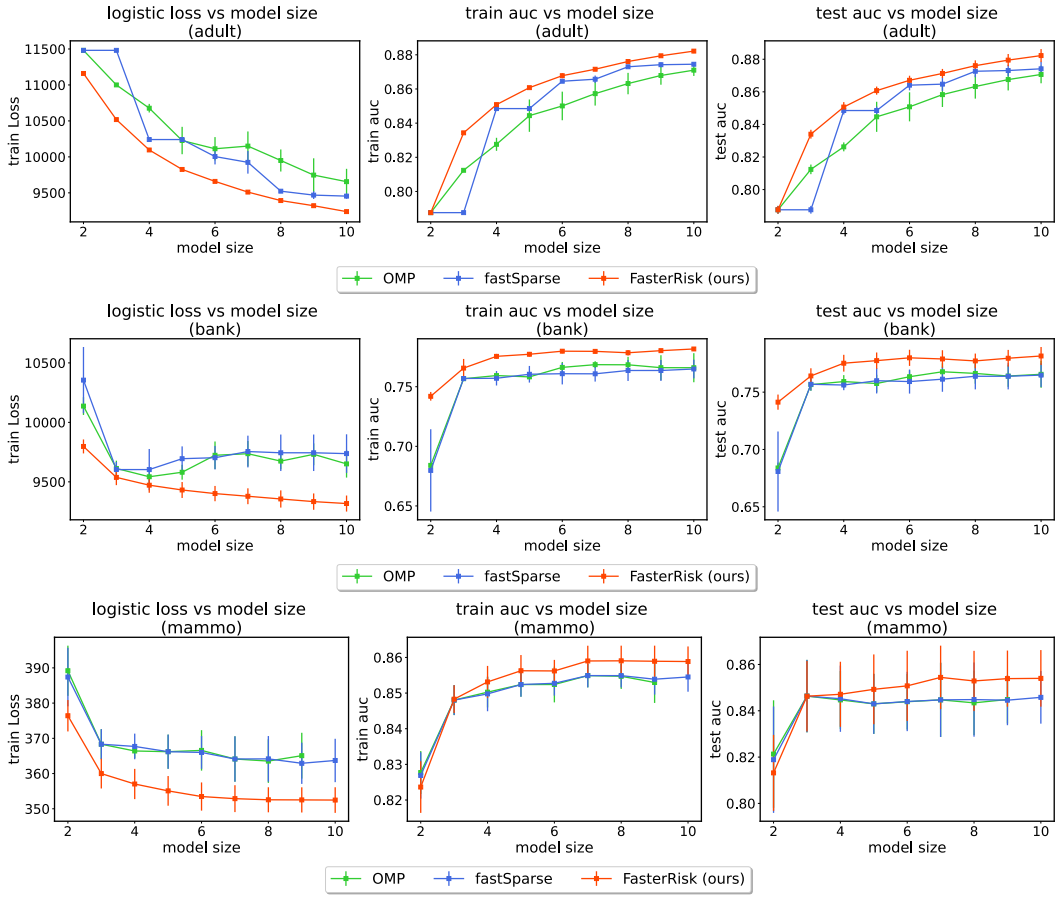

Figure 16: Sparse integer solutions on the adult, bank, and mammo datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). FasterRisk consistently outperforms the other two methods, due to higher quality of continuous sparse solutions and the use of multipliers.

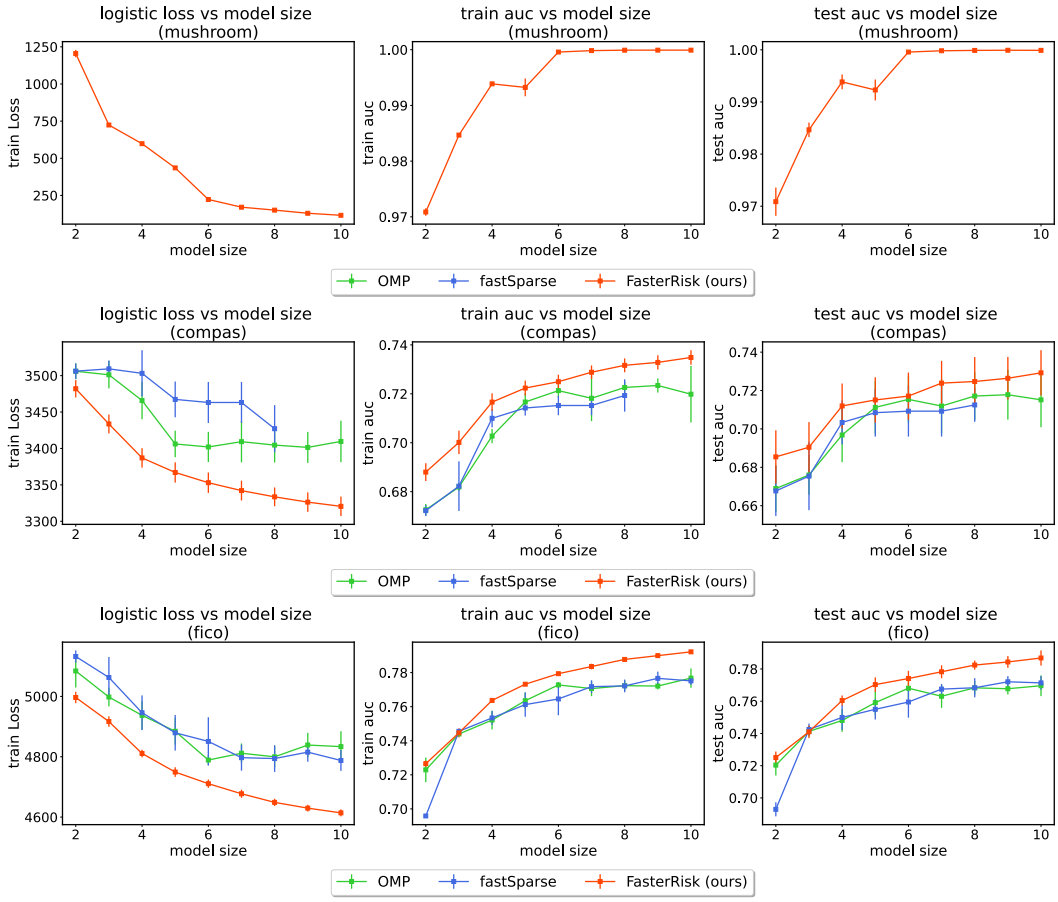

Figure 17: Sparse integer solutions on the mushroom, COMPAS, and FICO datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). The solution coefficients from the OMP and fastSparse methods violate the box constraints on the mushroom dataset, so we omit the results on the plot. fastSparse cannot obtain solutions with model size equal to 9 or 10 on the COMPAS dataset, so we do not show those points on the plot.

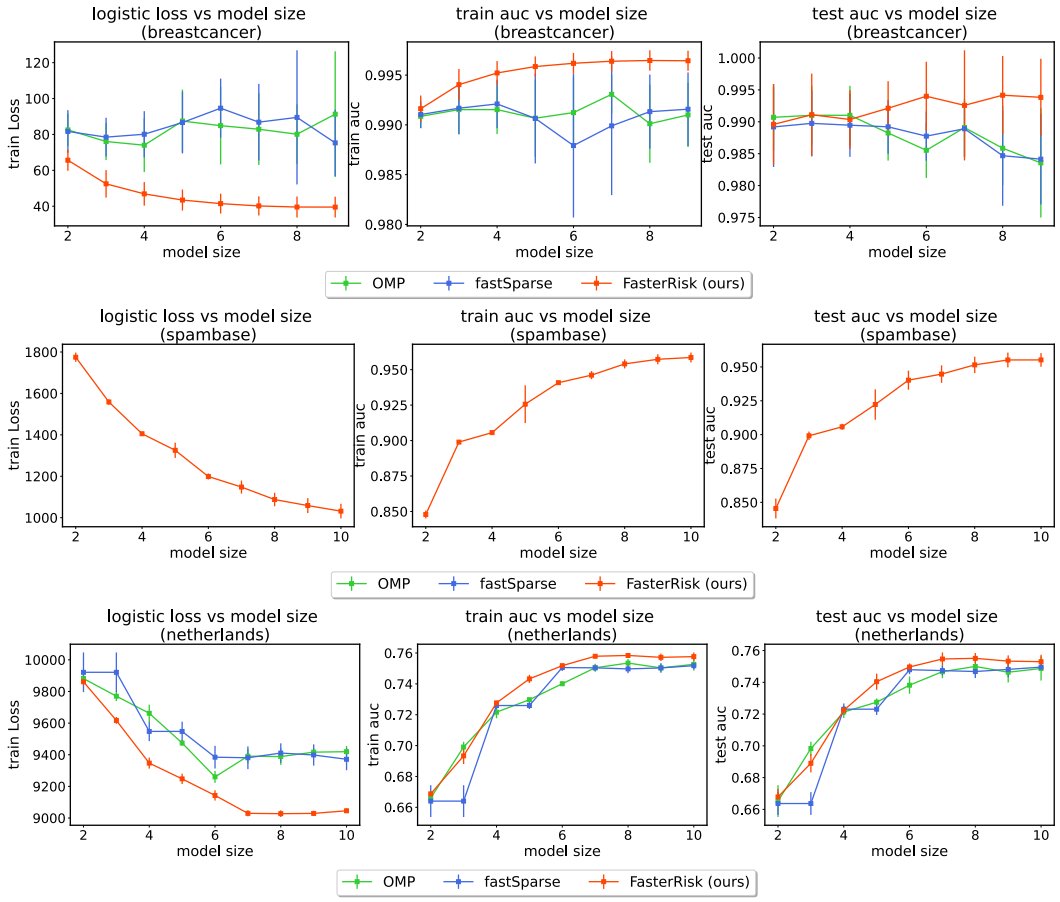

Figure 18: Sparse integer solutions on the breastcancer, spambase, and Netherlands datasets. Left column is loss (lower is better), middle column is training AUC (higher is better) and right column is test AUC (higher is better). FasterRisk consistently outperforms the other two methods, due to the higher quality of the continuous sparse solutions and the use of multipliers. The solution coefficients by the OMP and fastSparse methods violate the box constraints on the spambase dataset, so we omit the results in the plot.

E.8 Running RiskSLIM Longer

The experiments in Section 4 imposed a 900-second timeout, and RiskSLIM frequently did not complete within the 900 seconds. Here, we run RiskSLIM with longer timeouts (1 hour, and 4 days). We find that even with these long runtimes, FasterRisk still outperforms RiskSLIM in both solution quality and runtime.

Runtime is important for two reasons: (1) We may not be able to compute the answer at all using the slow method because it does not scale to reasonably-sized datasets. It could take a week or more to compute the solution for even reasonably small datasets. We will show this shortly through experiments. (2) Machine learning in the wild is never a single run of an algorithm. Often, users want to explore the data and adjust various constraints as they become more familiar with possible models. A fast speed allows users to go through this iteration process many times without lengthy interruptions between runs. This is where FasterRisk will be very useful in high stakes offline settings. FasterRisk's pool of models is generated within 5 minutes, and interacting with the pool is essentially instantaneous after it is generated.

E.8.1 Solution Quality of Running RiskSLIM for 1 hour

We ran RiskSLIM for a time limit of 1 hour on all 5 folds and all model sizes (2-10). Thus, we ran experiments for 2 days per dataset. As a reminder, our method FasterRisk runs in less than 5 minutes (on all datasets). The results of logistic loss on the training set, AUC on the training set, and AUC on the test set are in Figures 19-21. FasterRisk still outperforms RiskSLIM in almost all cases, because it uses a larger search space.

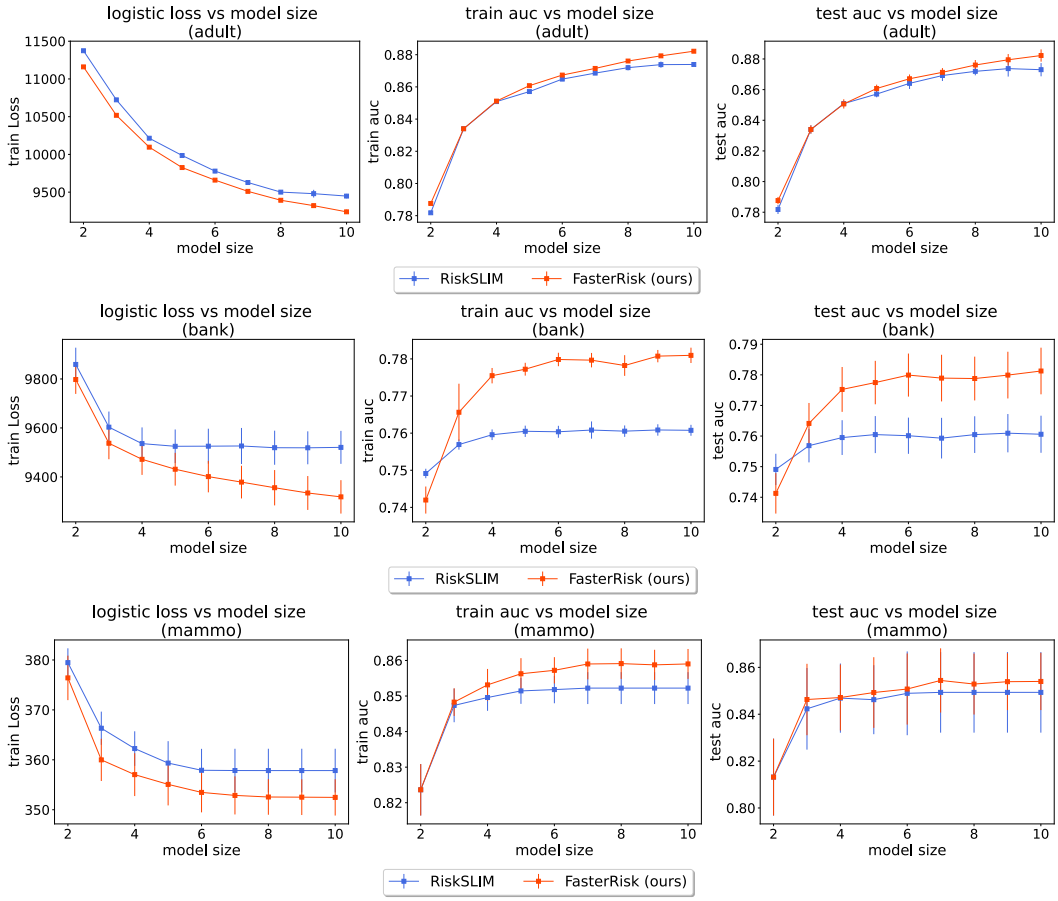

Figure 19: Comparison with the state-of-the-art baseline RiskSLIM (running for 1 hour) on the adult, bank, and mammo datasets. The left column is loss (lower is better), the middle column is training AUC (higher is better) and the right column is test AUC (higher is better).

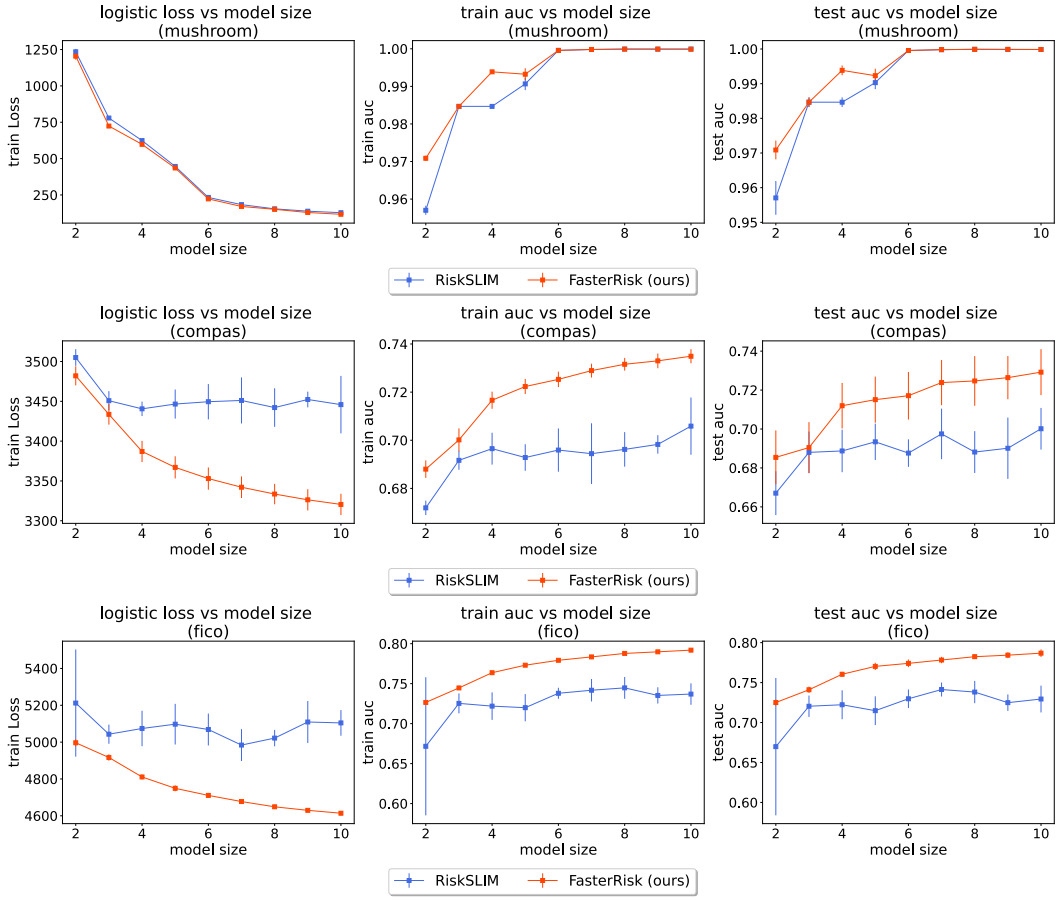

Figure 20: Comparison with the state-of-the-art baseline RiskSLIM (running for 1 hour) on the mushroom, COMPAS, and FICO datasets. The left column is loss (lower is better), the middle column is training AUC (higher is better) and the right column is test AUC (higher is better).

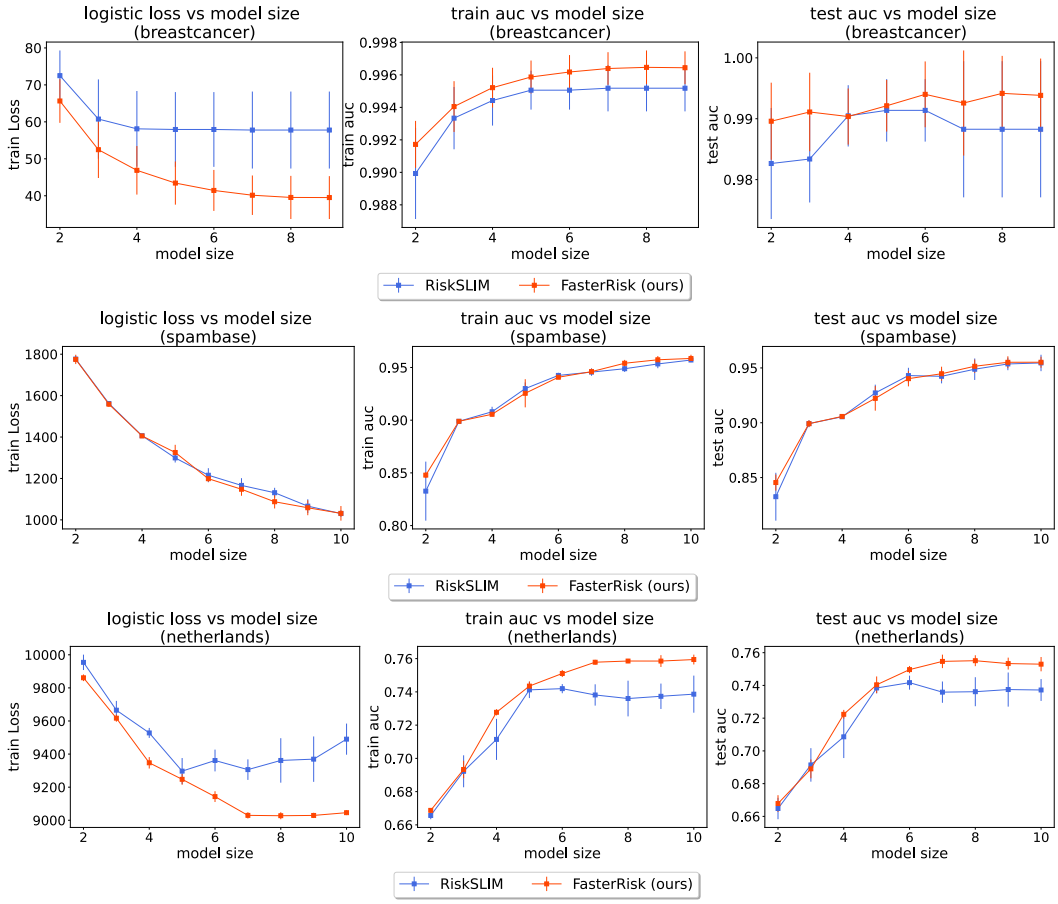

Figure 21: Comparison with the state-of-the-art baseline RiskSLIM (running for 1h) on the breast-cancer, spambase, and Netherlands datasets. The left column is loss (lower is better), the middle column is training AUC (higher is better) and the right column is test AUC (higher is better).

E.8.2 Time Comparison of Running RiskSLIM for 1 hour

We plot the running time comparison between FasterRisk and RiskSLIM (with a time limit of 1 hour). The original time results with the 15-minute time limit are shown in Figure 5 and Figure 8.

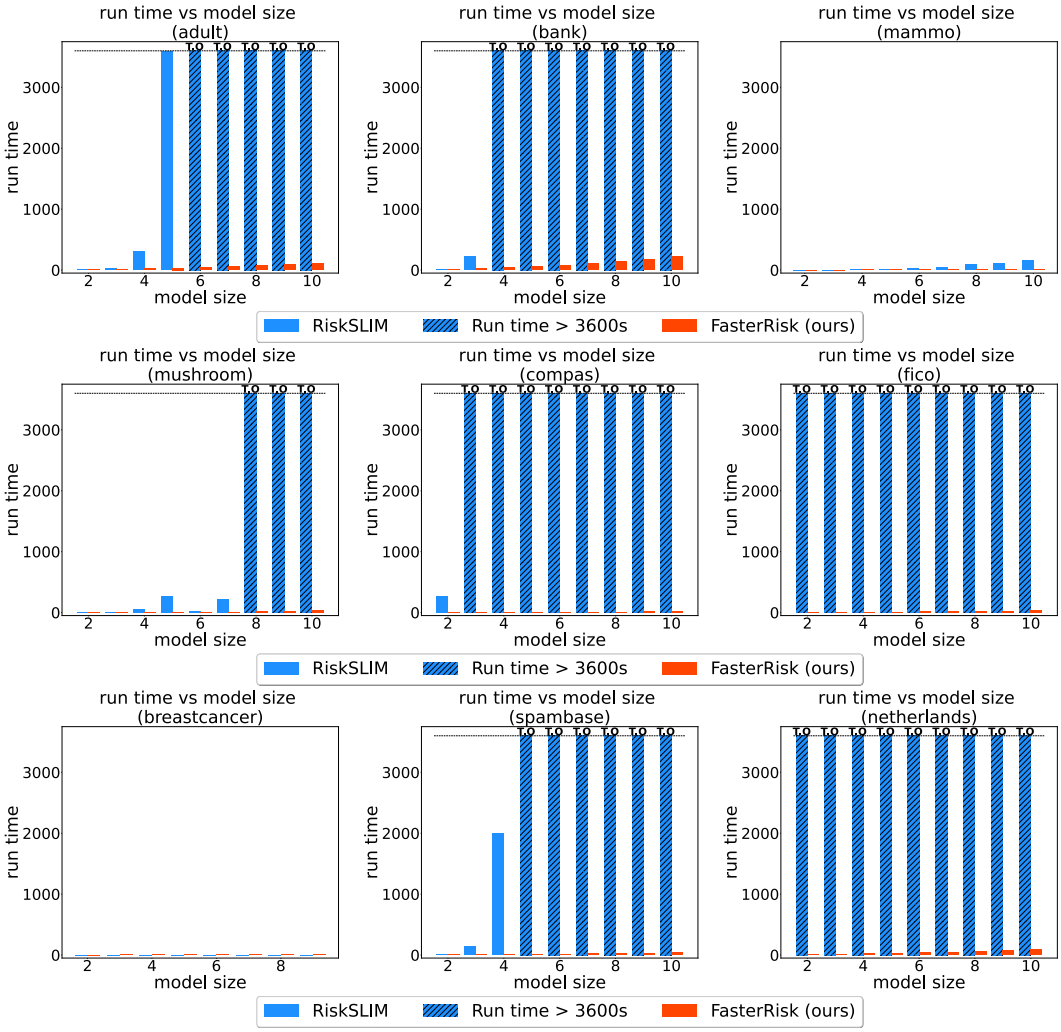

Figure 22: Runtime Comparison. Runtime (in seconds) versus model size for our method FasterRisk (in red) and the RiskSLIM (in blue). The shaded blue bars indicate cases that timed out at 1 hour. Breastcancer is a small dataset so it takes approximately 2 seconds for both algorithms. For more zoomed-in results on the breastcancer and mammo datasets, please refer to Figure 5 and Figure 8.

E.8.3 Solution Quality of Running RiskSLIM for Days

We report results of running the baseline RiskSLIM for 4 days. Due to this long running time demand on our servers, we could not run this experiment on all folds and all model sizes, so we only run on the 3rd fold of the 5-CV split. We plot the logistic loss progression over time.

The results are shown in Figure 23. We see that FasterRisk still achieves lower loss than RiskSLIM even after letting RiskSLIM run for 4 days, again because FasterRisk uses a larger model class. The only exceptions are on the Mushroom and the Spambase datasets, where the logistic losses are close to each other.

The major disadvantage of letting an algorithm run for days is that it is challenging to interact with the algorithm, because one has to wait for the results between interactions – ideally this process would be instantaneous. Furthermore, there could be memory issues for the MIP solver if we let it run for days since the branch-and-bound tree could become too large.

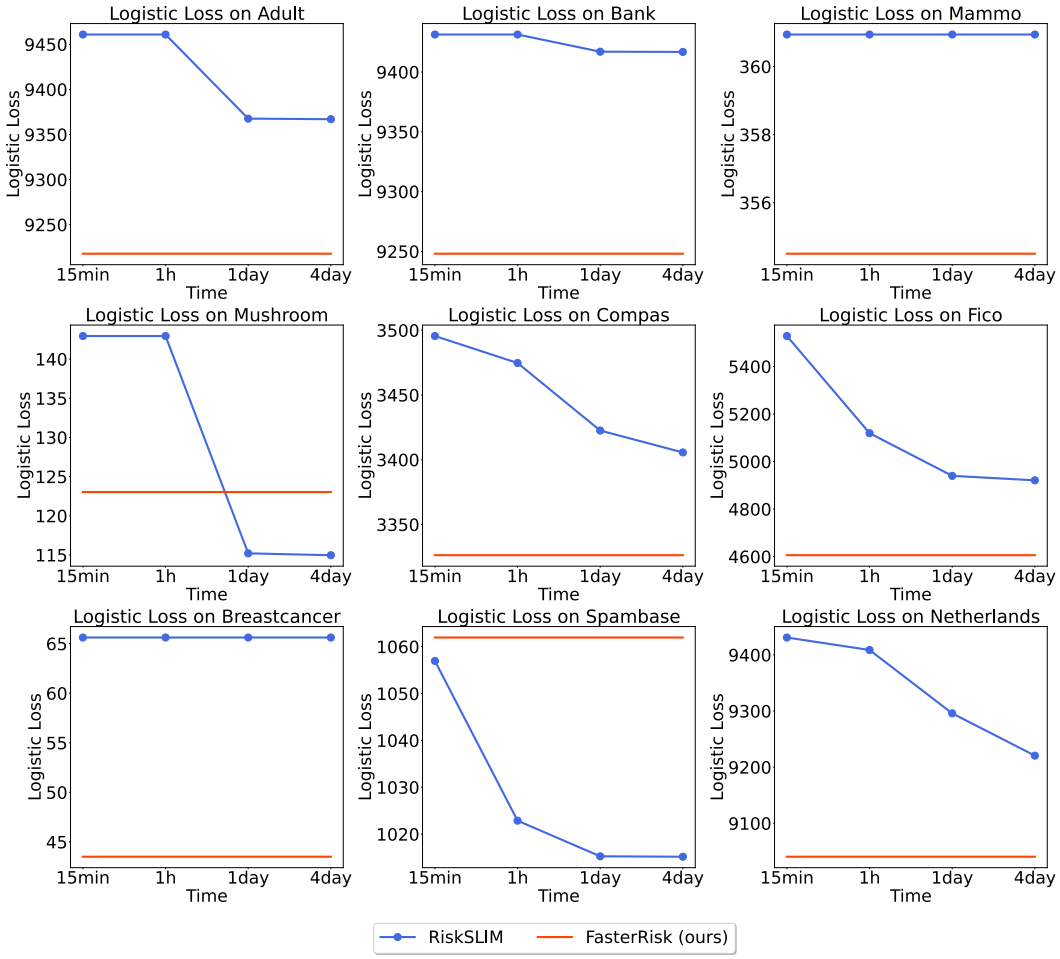

Figure 23: Curves of logistic loss vs. training time for the RiskSLIM model on the 3rd fold of the 5-CV split with model size equal to 10. All plots report logistic loss (lower is better).

E.9 Calibration Curves

The calibration curves for RiskSLIM and FasterRisk are shown in Figures 24-26 with model sizes equal to 3, 5, and 7, respectively. We use the sklearn package[4] from python to plot the figures. We use the default value for the number of bins (number of bins is 5) and the default strategy to define the widths of the bins (the strategy is "uniform").

The calibration curves on the breastcancer and mammo datasets are more spread out than those on the other datasets. This is perhaps due to the limited number of samples in these datasets (both datasets have fewer than 1000 samples in total; see Table 2), which increases the variance in the calculation of the curves.

On other datasets, both methods have good calibration curves, showing consistency between predicted score and actual risk. However, as shown in Figures 19-21, FasterRisk has higher AUC scores, which means our method has higher discrimination ability than RiskSLIM.

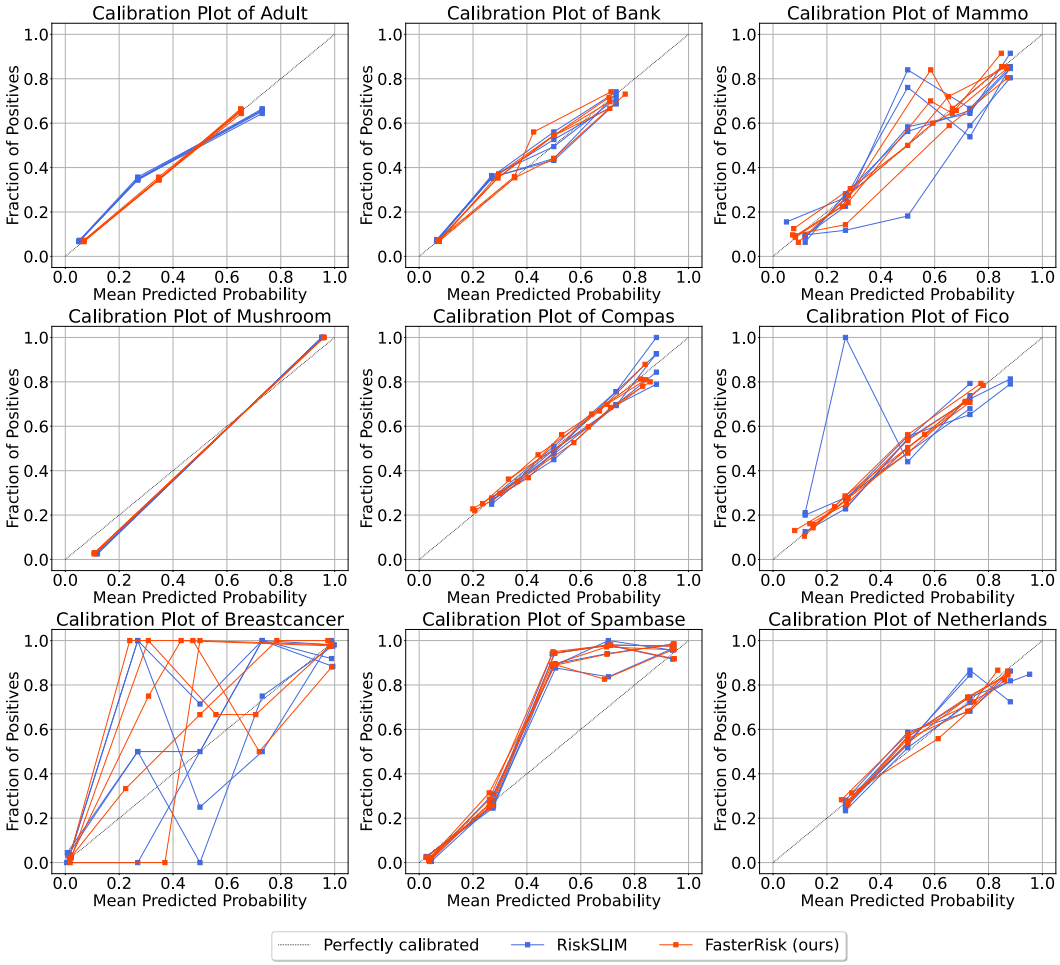

Figure 24: Calibration curves for RiskSLIM and FasterRisk with model size equal to 3. We plot results from each test fold. The FasterRisk model selected from the pool is that with the smallest logistic loss on the training set.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.calibration.calibration_curve.html

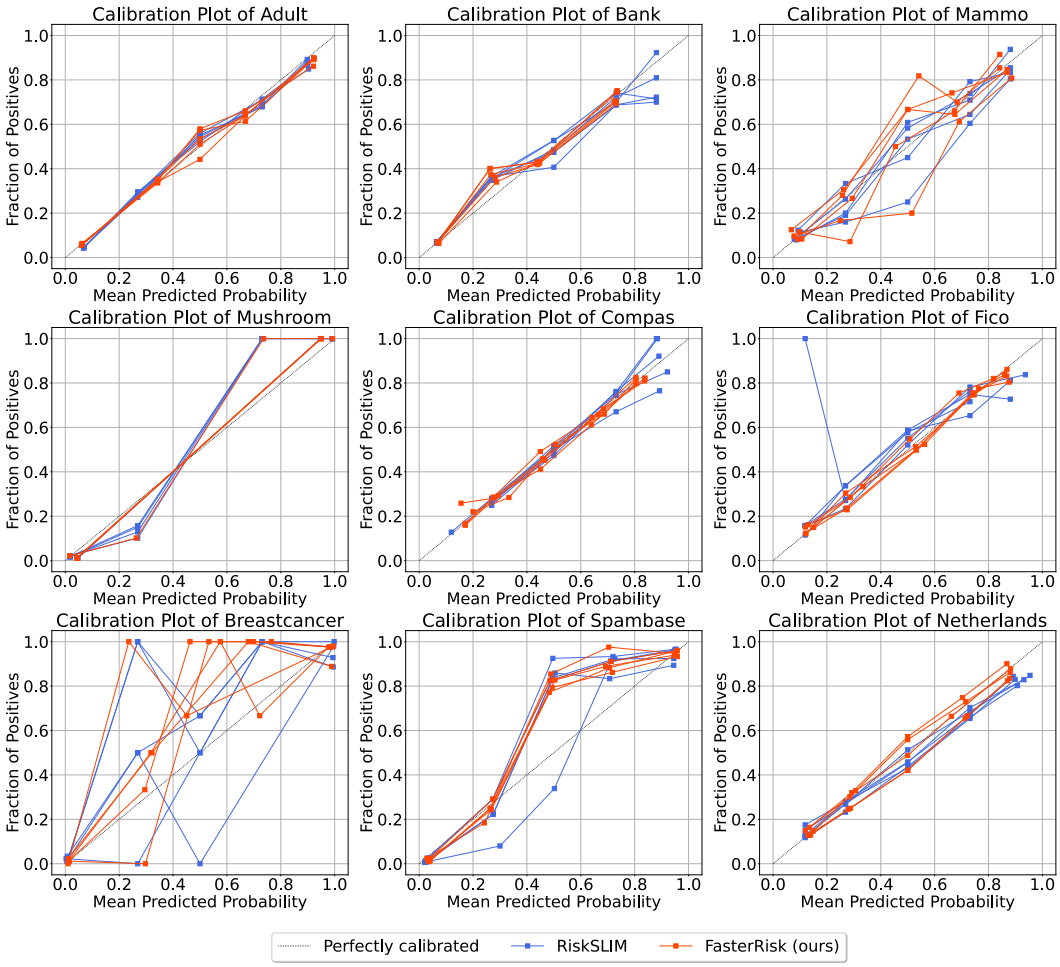

Figure 25: Calibration curves for RiskSLIM and FasterRisk with model size equal to 5. We plot results from each test fold. The FasterRisk model selected from the pool is that with the smallest logistic loss on the training set.

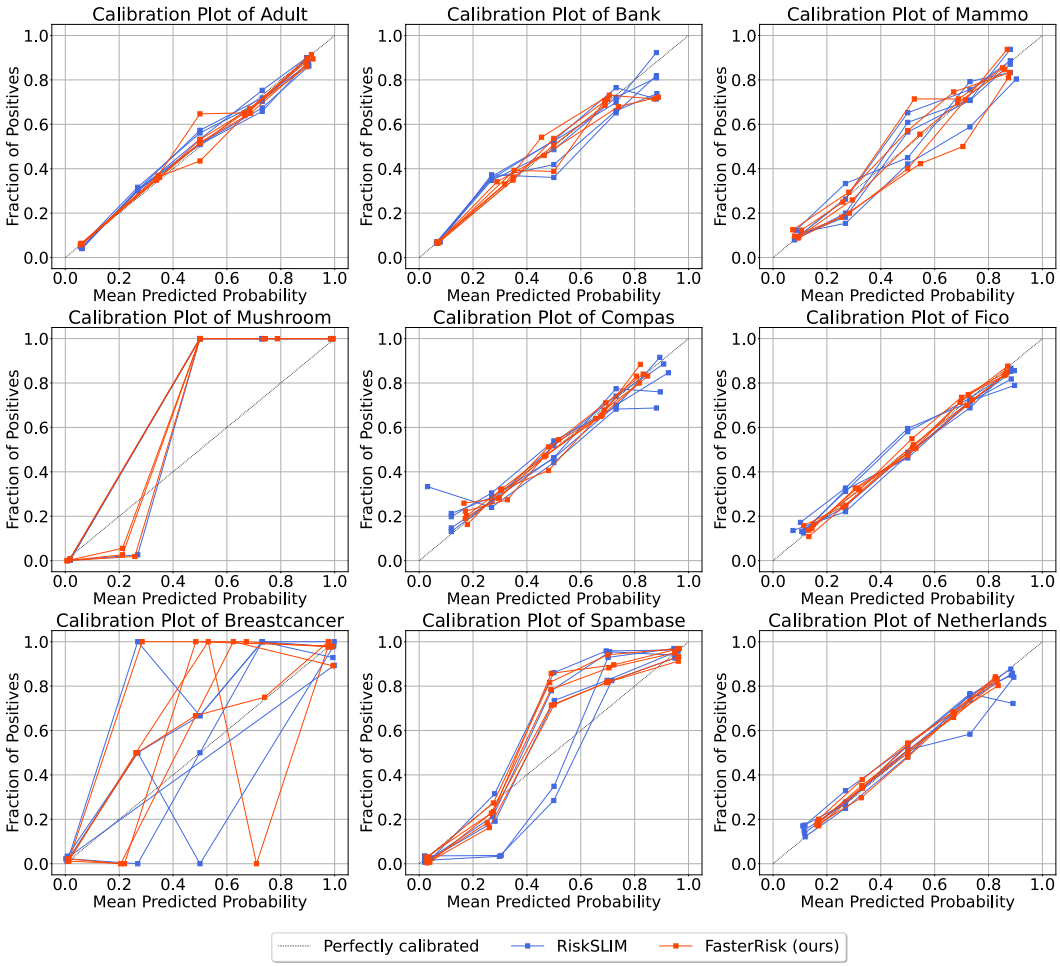

Figure 26: Calibration curves for RiskSLIM and FasterRisk with model size equal to 7. We plot results from each fold on the test set. The FasterRisk model selected from the pool is that with the smallest logistic loss on the training set.

E.10 Hyperparameter Perturbation Study

E.10.1 Perturbation Study on Beam Size B

We perform a perturbation study on the hyperparameter beam size B as mentioned in Appendix D.4. We set the beam size to 5, 10, and 15, respectively. The results are shown in Figures 27-29. The curves greatly overlap, confirming our previous claim that the performance is not particularly sensitive to the choice of B.

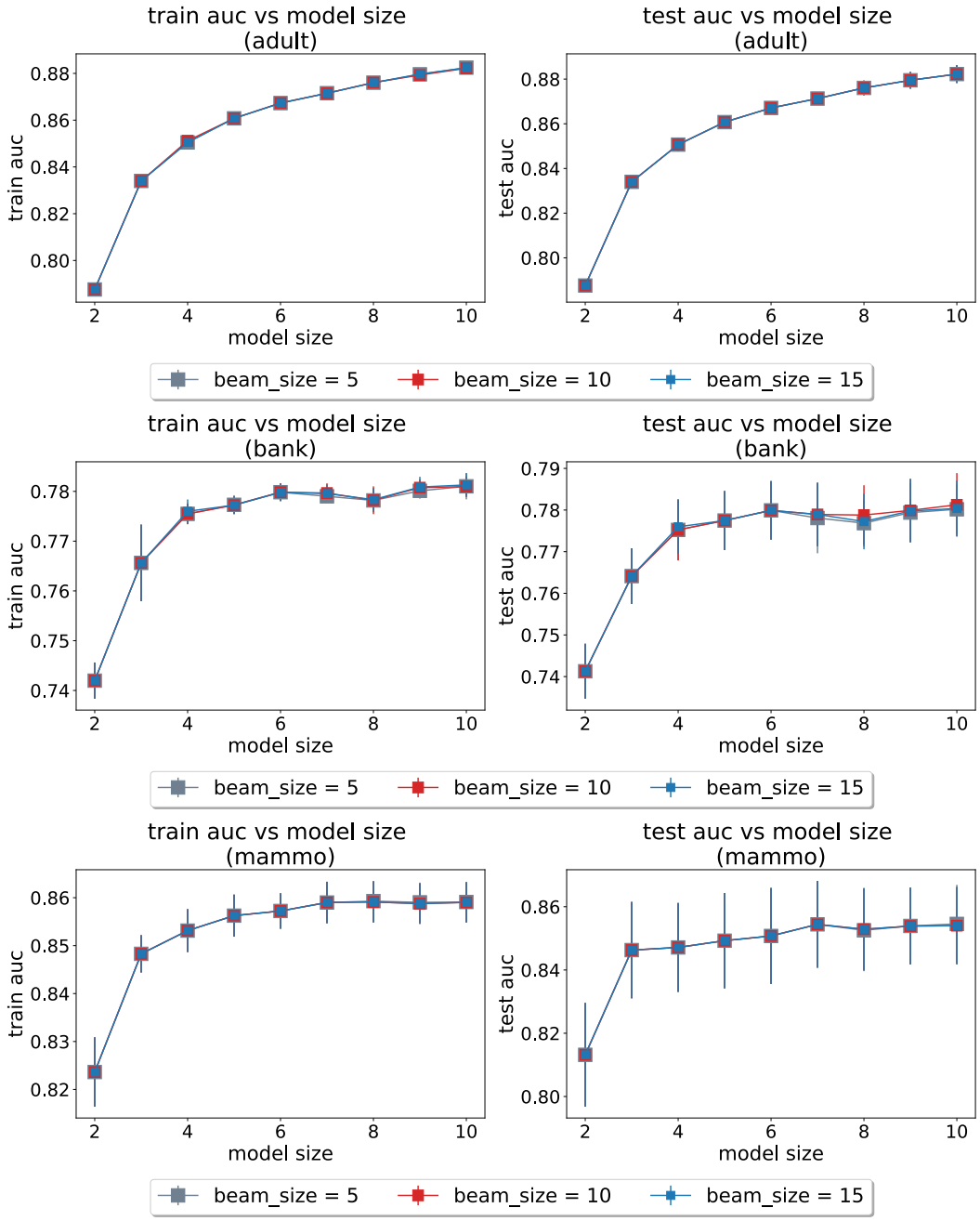

Figure 27: Perturbation study for beam size, B, on the adult, bank, and mammo datasets. The default value used in the paper is 10.

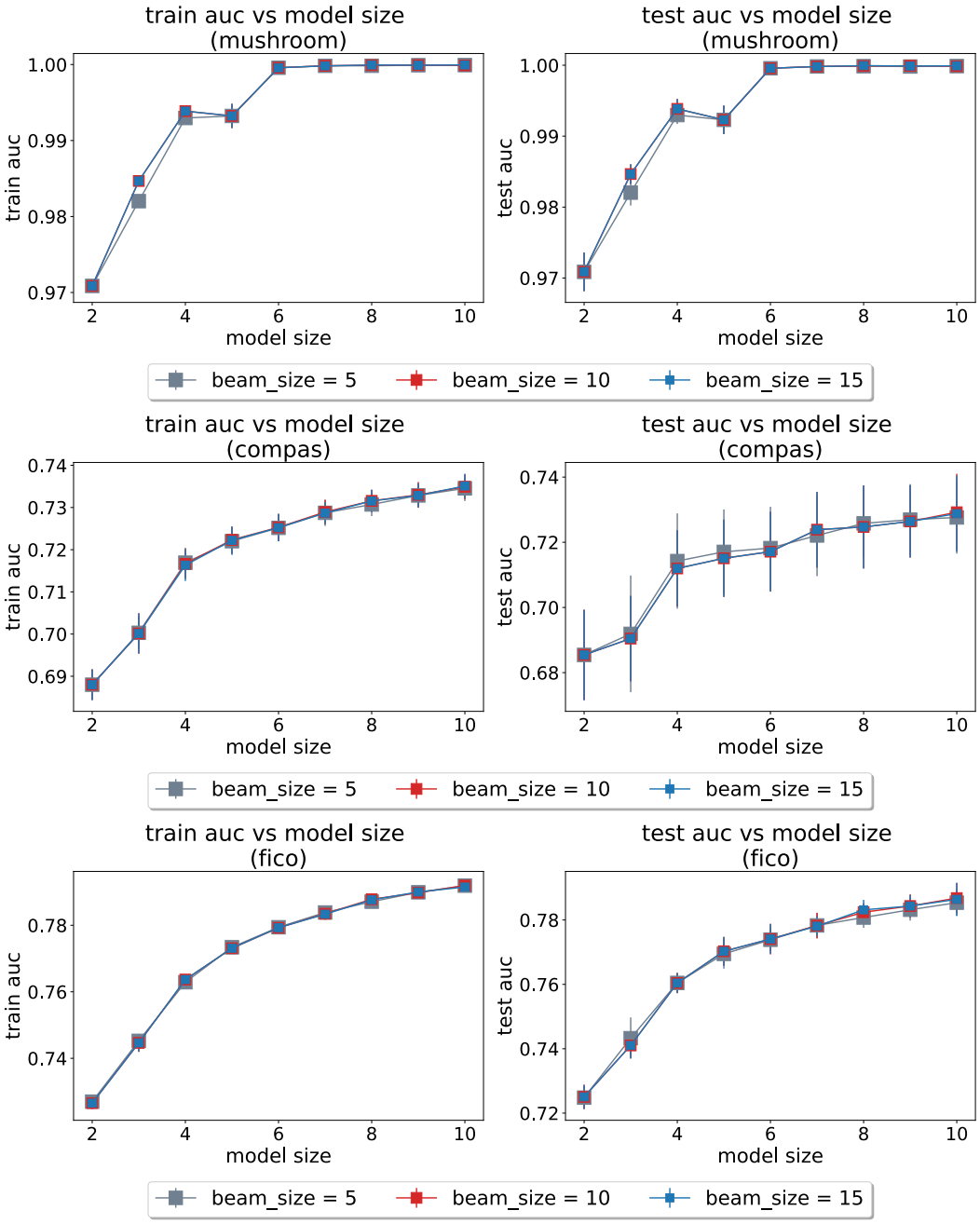

Figure 28: Perturbation study for beam size, B, on the mushroom, COMPAS, and FICO datasets. The default value used in the paper is 10.

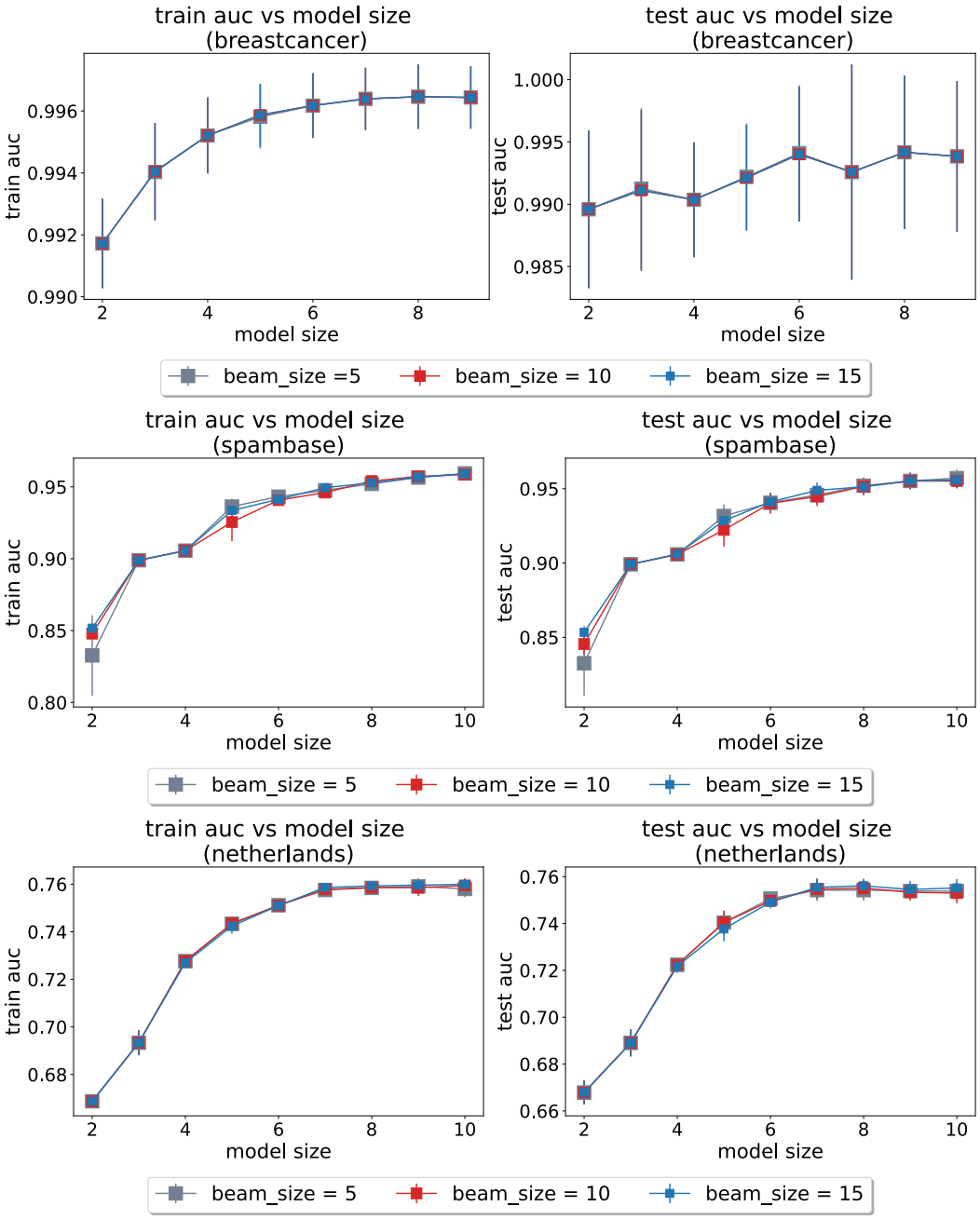

Figure 29: Perturbation study for beam size, B, on the breastcancer, spambase, and Netherlands datasets. The default value used in the paper is 10.

E.10.2 Perturbation Study on Tolerance Level ϵ for Sparse Diverse Pool

We perform a perturbation study on the hyperparameter tolerance level, ϵ, as mentioned in Appendix D.4. We set the tolerance level to 0.1, 0.3, and 0.5, respectively. The results are shown in Figures 30-32. The curves greatly overlap, confirming our previous claim that the performance is not particularly sensitive to the choice of value.

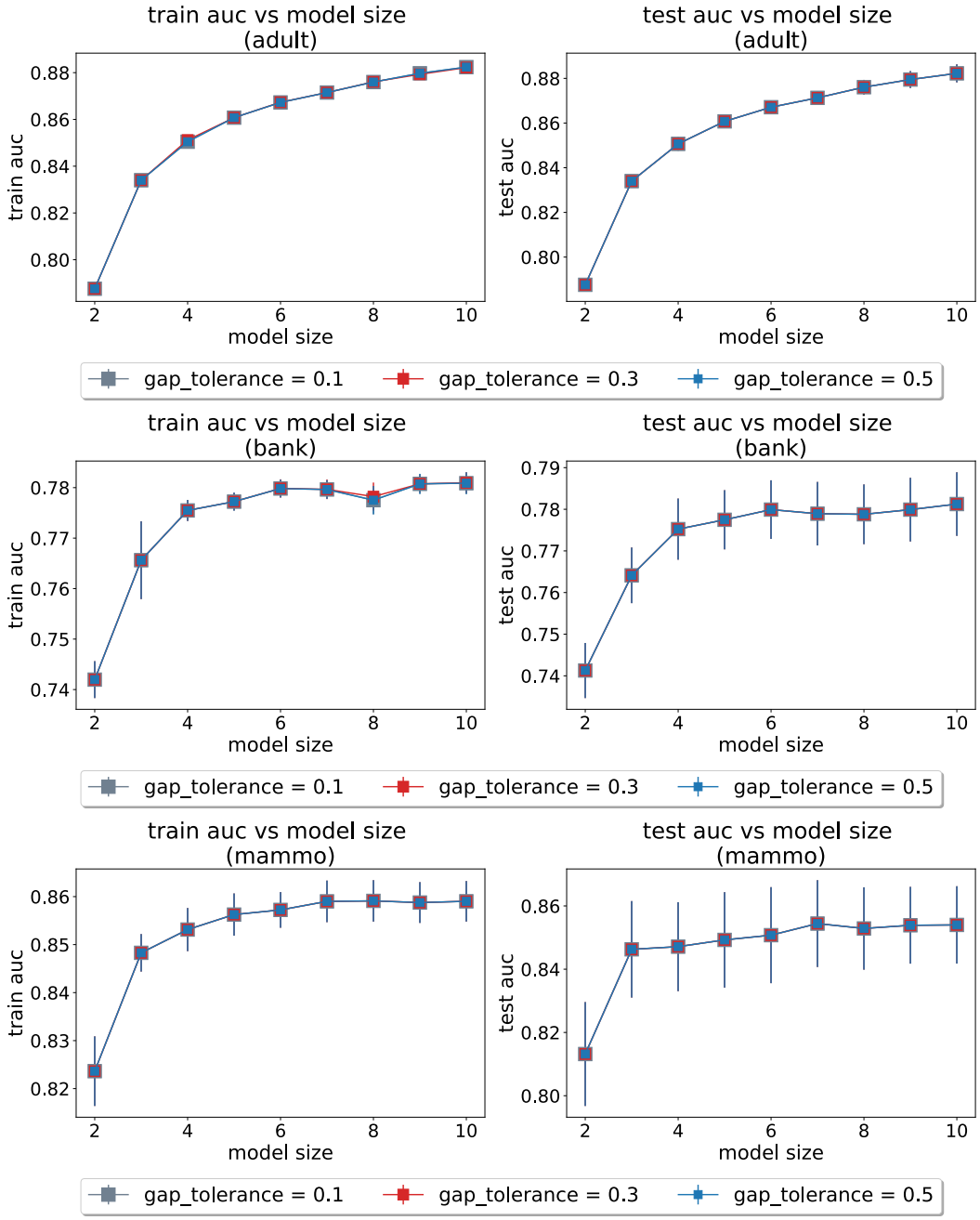

Figure 30: Perturbation study on tolerance level, ϵ, for sparse diverse pool on the adult, bank, and mammo datasets. The default value used in the paper is 0.3.

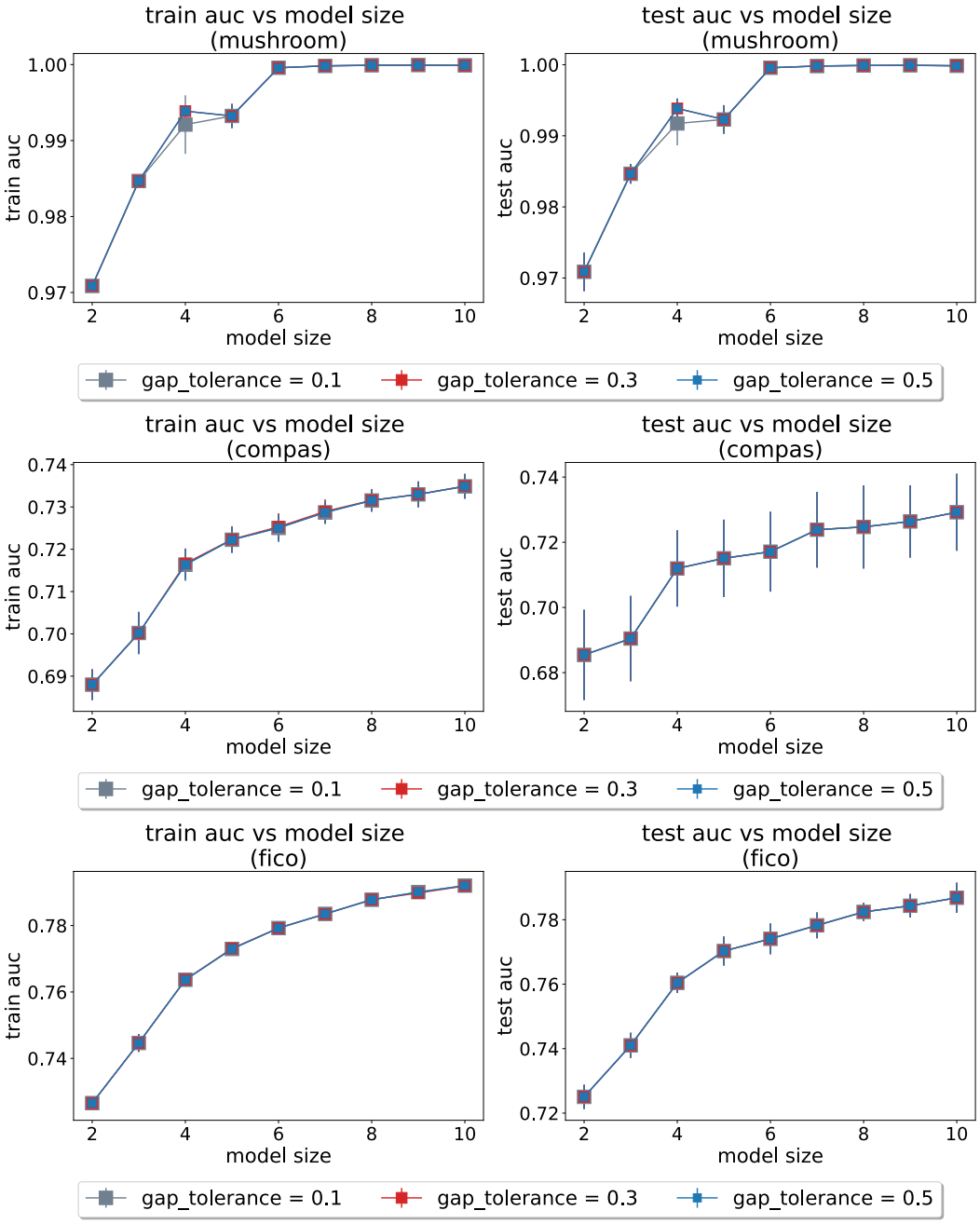

Figure 31: Perturbation study on tolerance level, ϵ, for sparse diverse pool on the mushroom, COMPAS and FICO datasets. The default value used in the paper is 0.3.

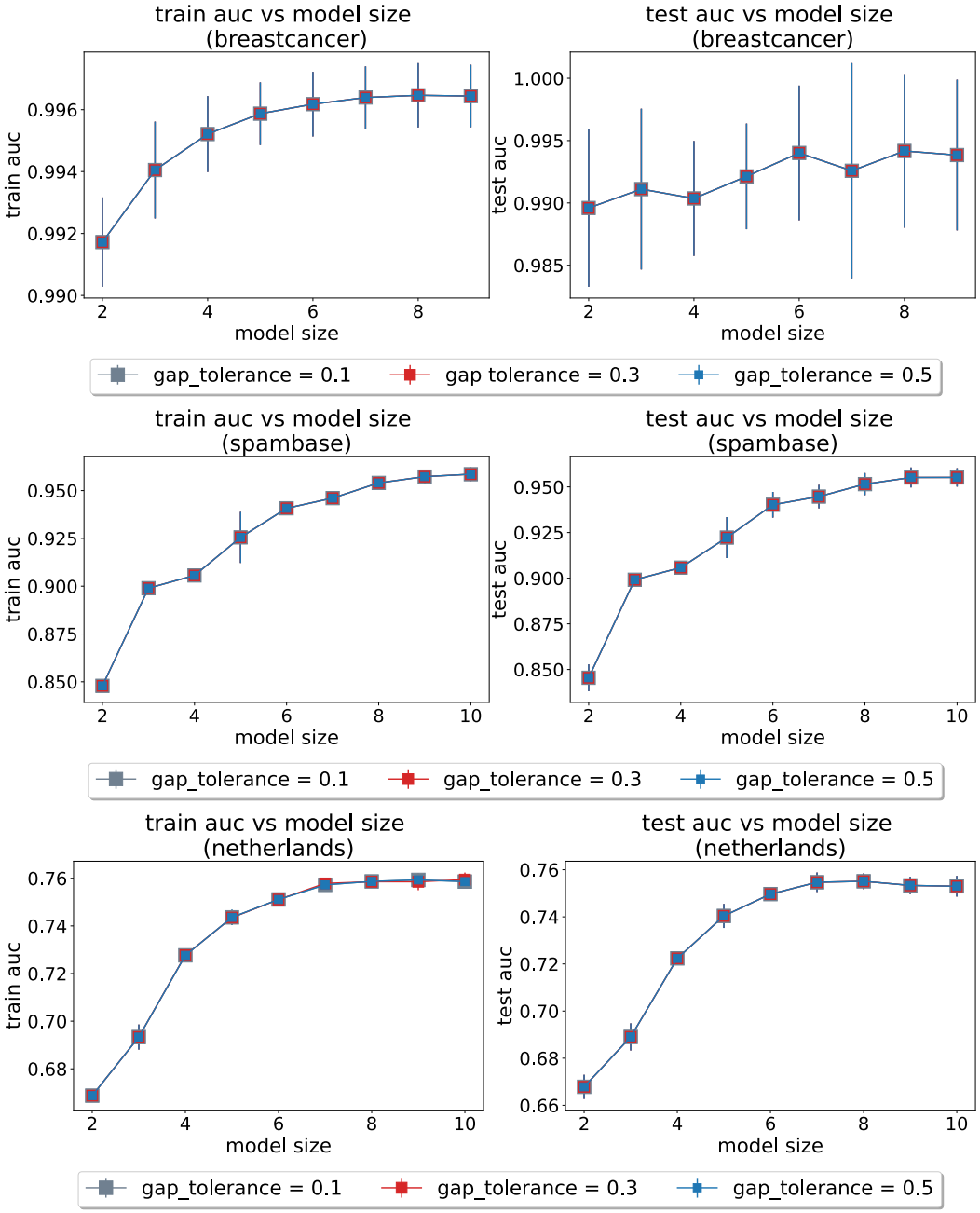

Figure 32: Perturbation study on tolerance level, ϵ, for sparse diverse pool on the breastcancer, spambase, and Netherlands datasets. The default value used in the paper is 0.3.

E.10.3 Perturbation Study on Number of Attempts T for Sparse Diverse Pool

We perform a perturbation study on the hyperparameter for the number of attempts, T, as mentioned in Appendix D.4. We have set the number of attempts to 35, 50, and 65, respectively. The results are shown in Figures 33-35. The curves greatly overlap, confirming our previous claim that the performance is not particularly sensitive to the choice of value for the hyperparameter.

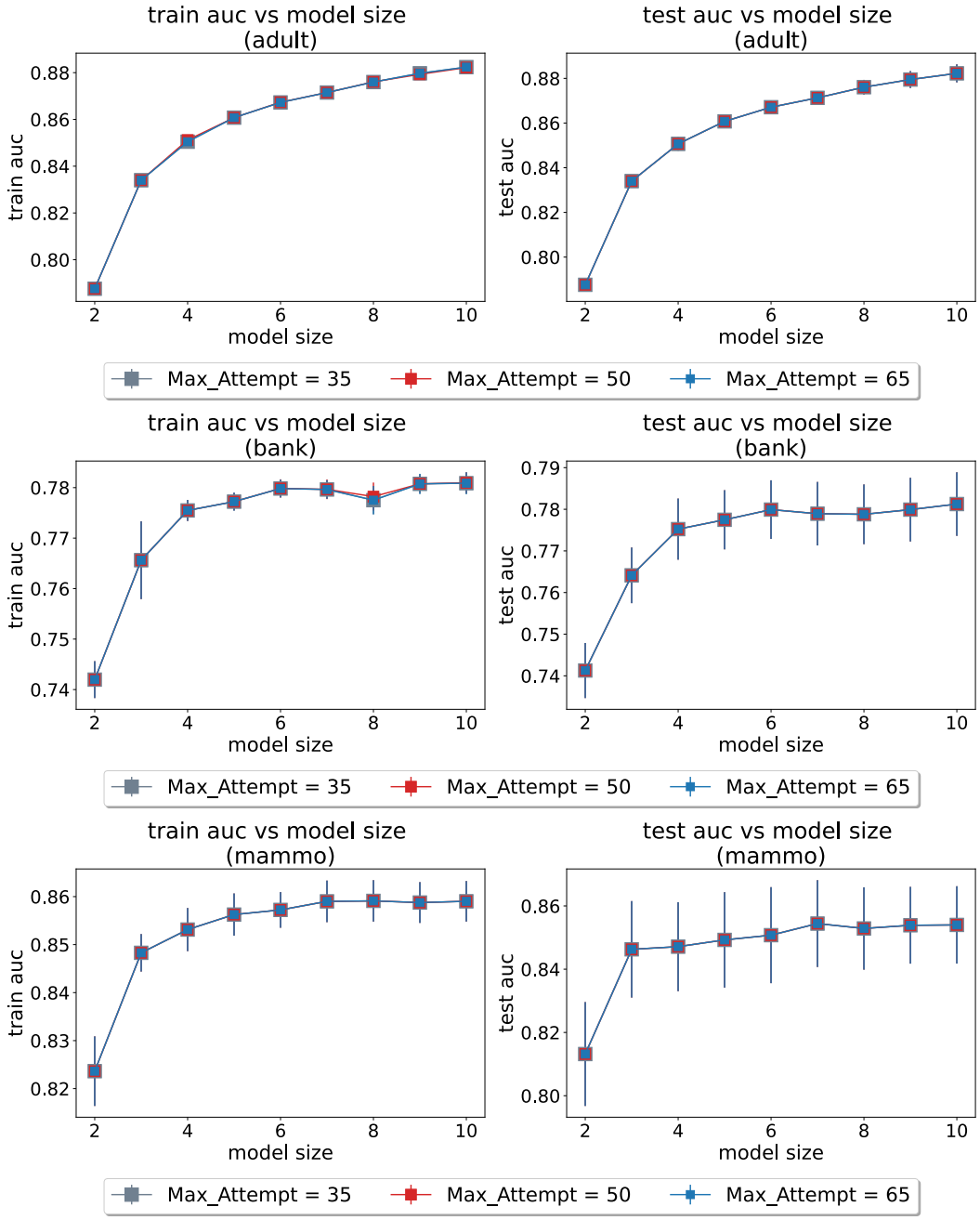

Figure 33: Perturbation study on number of attempts parameter, T, for sparse diverse pool on the adult, bank, and mammo datasets. The default value used in the paper is 50.

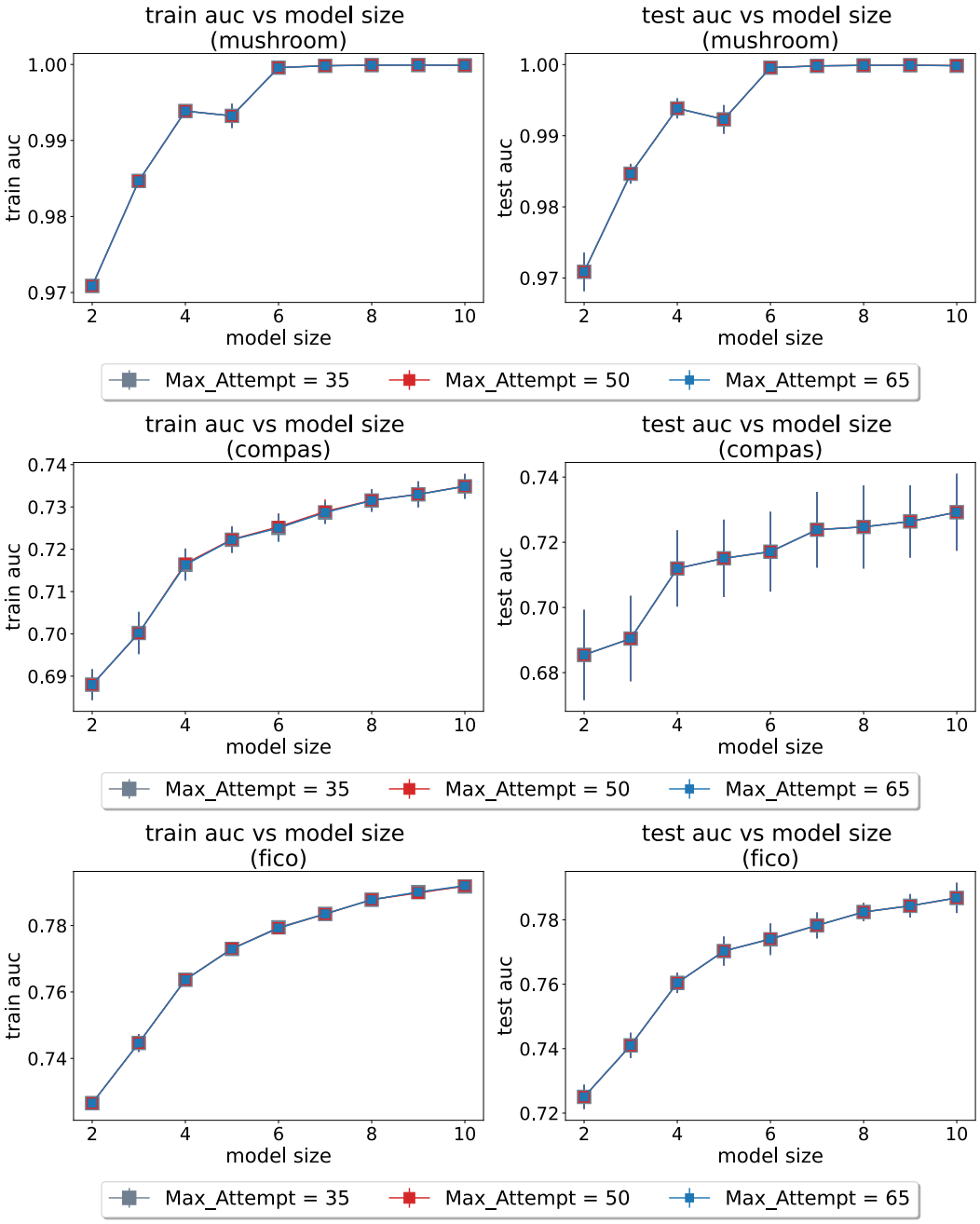

Figure 34: Perturbation study on number of attempts parameter, T, for sparse diverse pool on the mushroom, COMPAS, and FICO datasets. The default value used in the paper is 50.

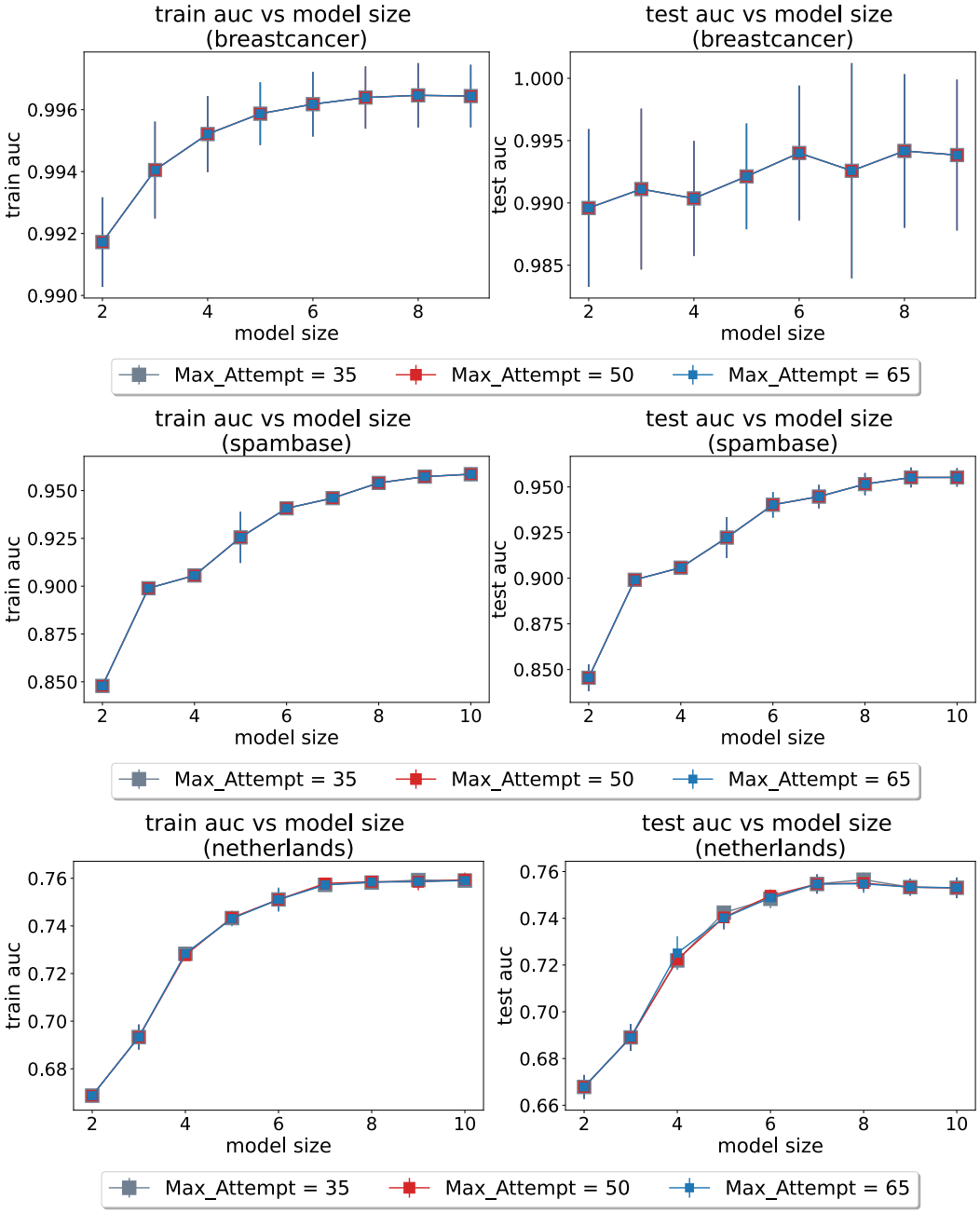

Figure 35: Perturbation study on number of attempts parameter, T, for sparse diverse pool on the breastcancer, spambase, and Netherlands datasets. The default value used in the paper is 50.

E.10.4 Perturbation Study on Number of Multipliers N_m

We perform a perturbation study on the hyperparameter for the number of multipliers, N_m, as mentioned in Appendix D.4. We have set the number of multipliers to 10, 20, and 30, respectively. The results are shown in Figures 36-38. The curves greatly overlap, confirming our previous claim that the performance is not particularly sensitive to the choice of values for N_m.

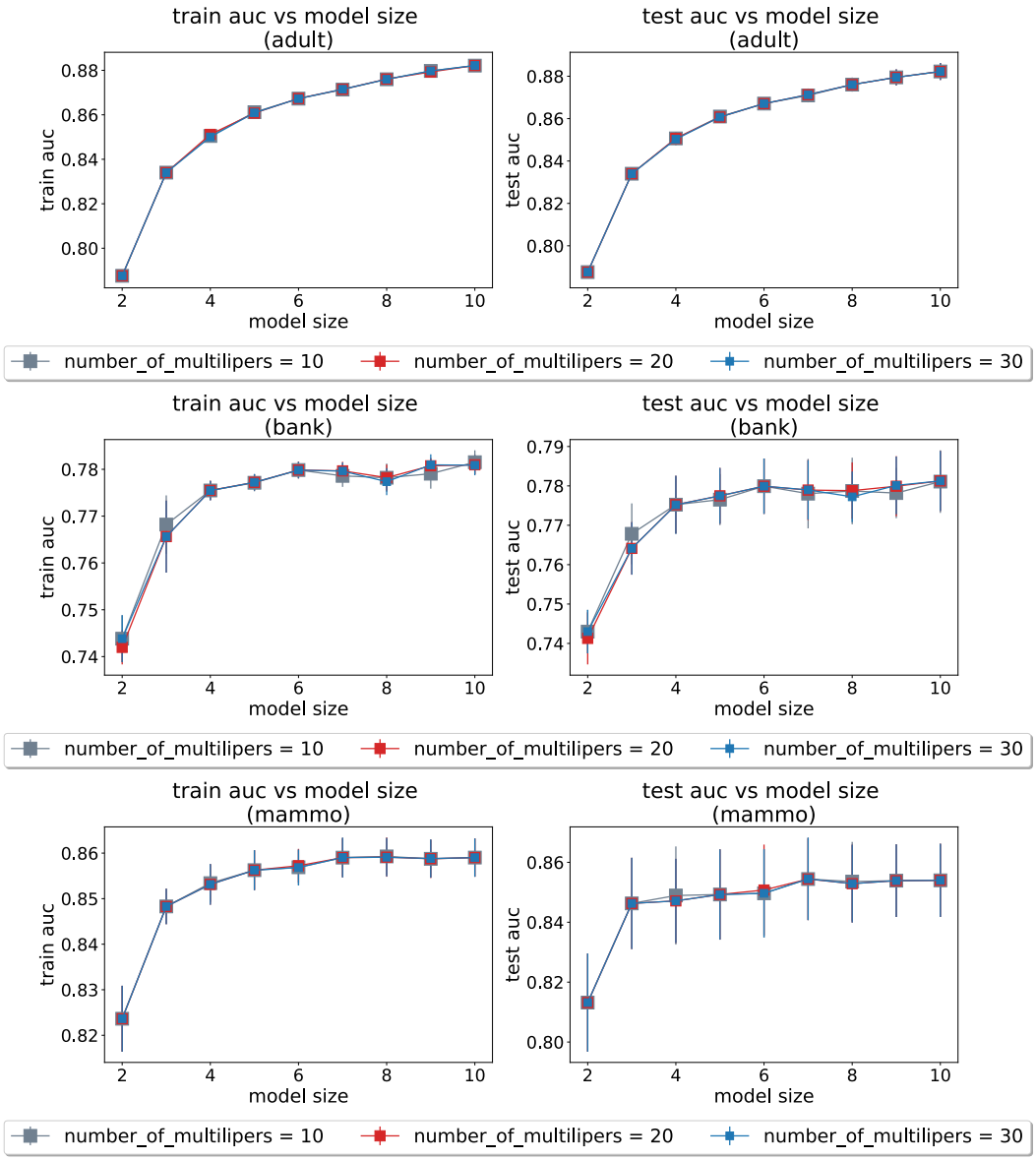

Figure 36: Perturbation study on number of multipliers, N_m, on the adult, bank, and mammo datasets. The default value used in the paper is 20.

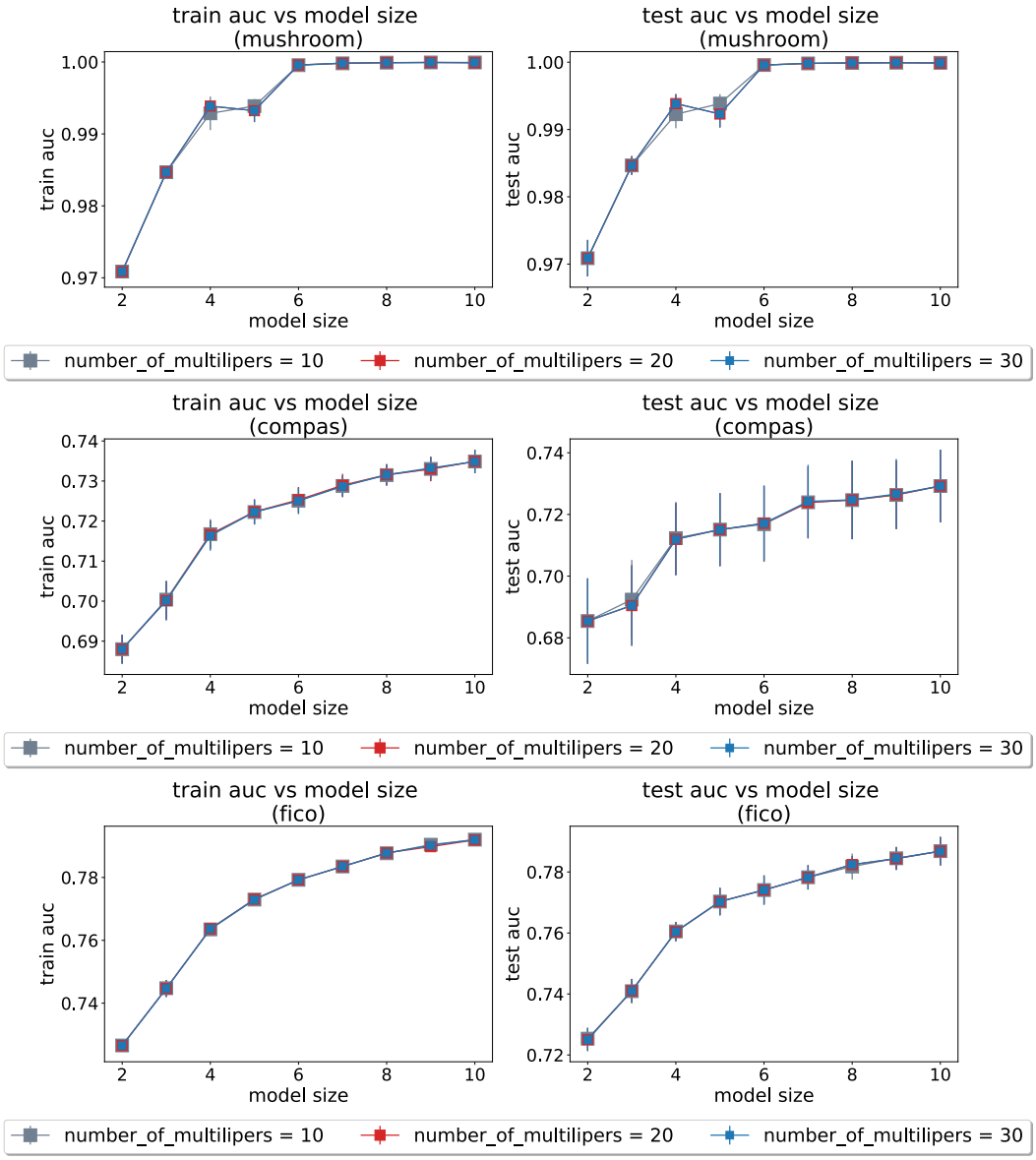

Figure 37: Perturbation study on number of multipliers, N_m, for sparse diverse pool on the mushroom, COMPAS and FICO datasets. The default value used in the paper is 20.

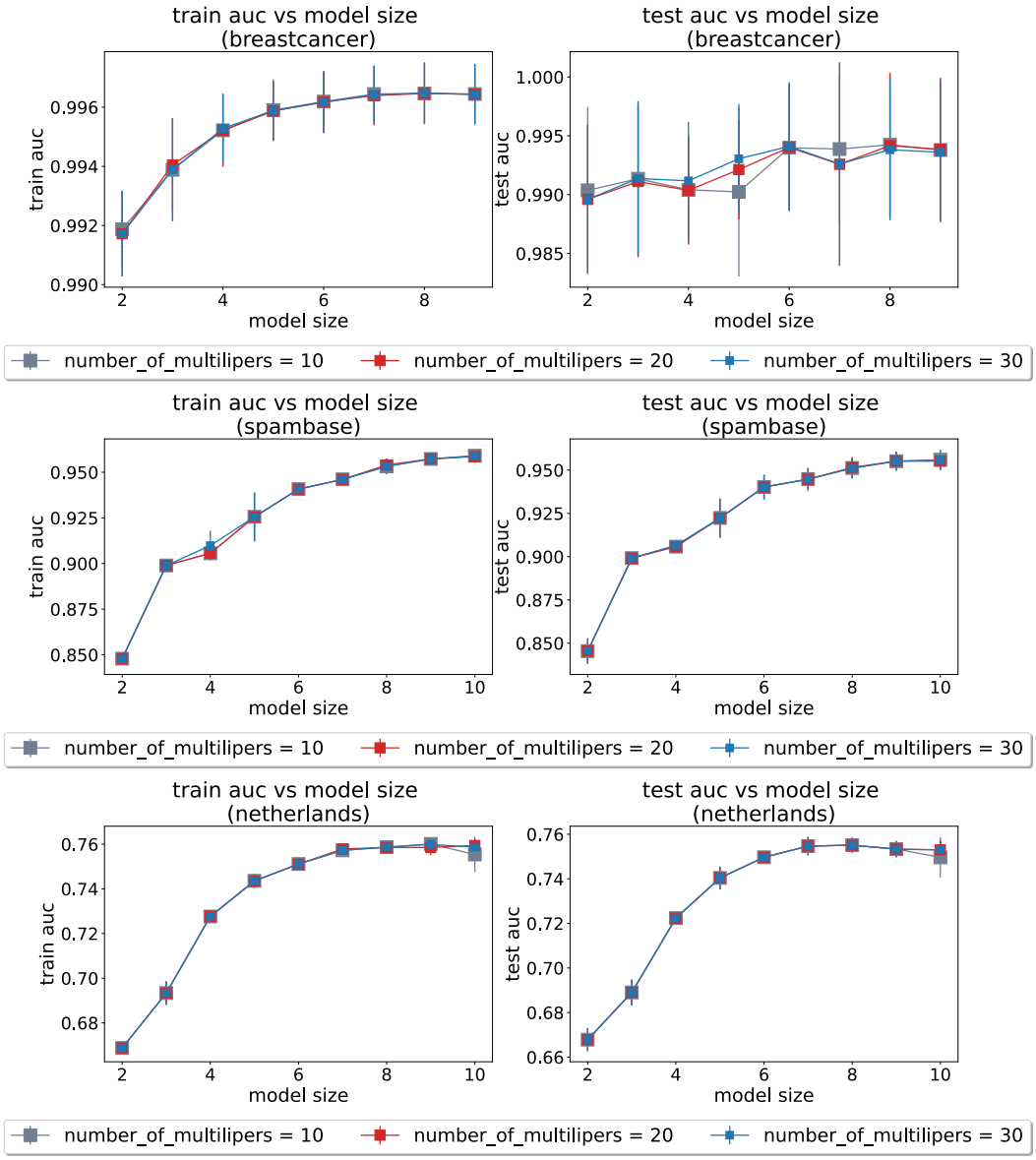

Figure 38: Perturbation study on number of multipliers, N_m, for sparse diverse pool on the breast-cancer, spambase, and Netherlands datasets. The default value used in the paper is 20.

E.11 Comparison with Baseline AutoScore

We compare with the baseline AutoScore [44]. We set the number of features from 2 to 10 and use all other hyperparameters in the default setting. The results of training AUC and test AUC are shown in Figures 39-41. The plots of RiskSLIM are from experiments where we let RiskSLIM run for 1 hour. FasterRisk outperforms both RiskSLIM and AutoScore.

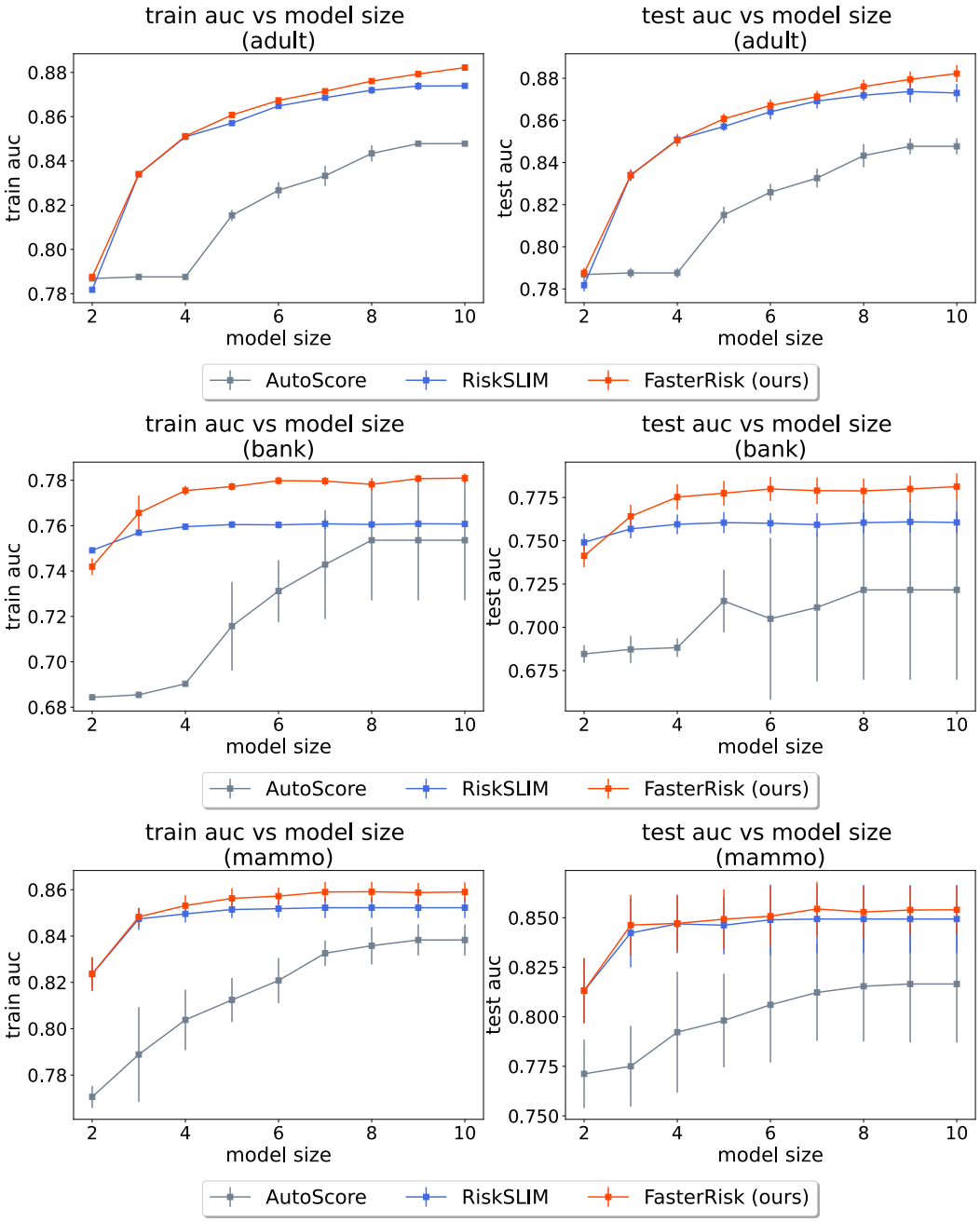

Figure 39: Comparison with the new baseline AutoScore on the adult, bank, and mammo datasets. The left column is training AUC (higher is better), and the right column is test AUC (higher is better).

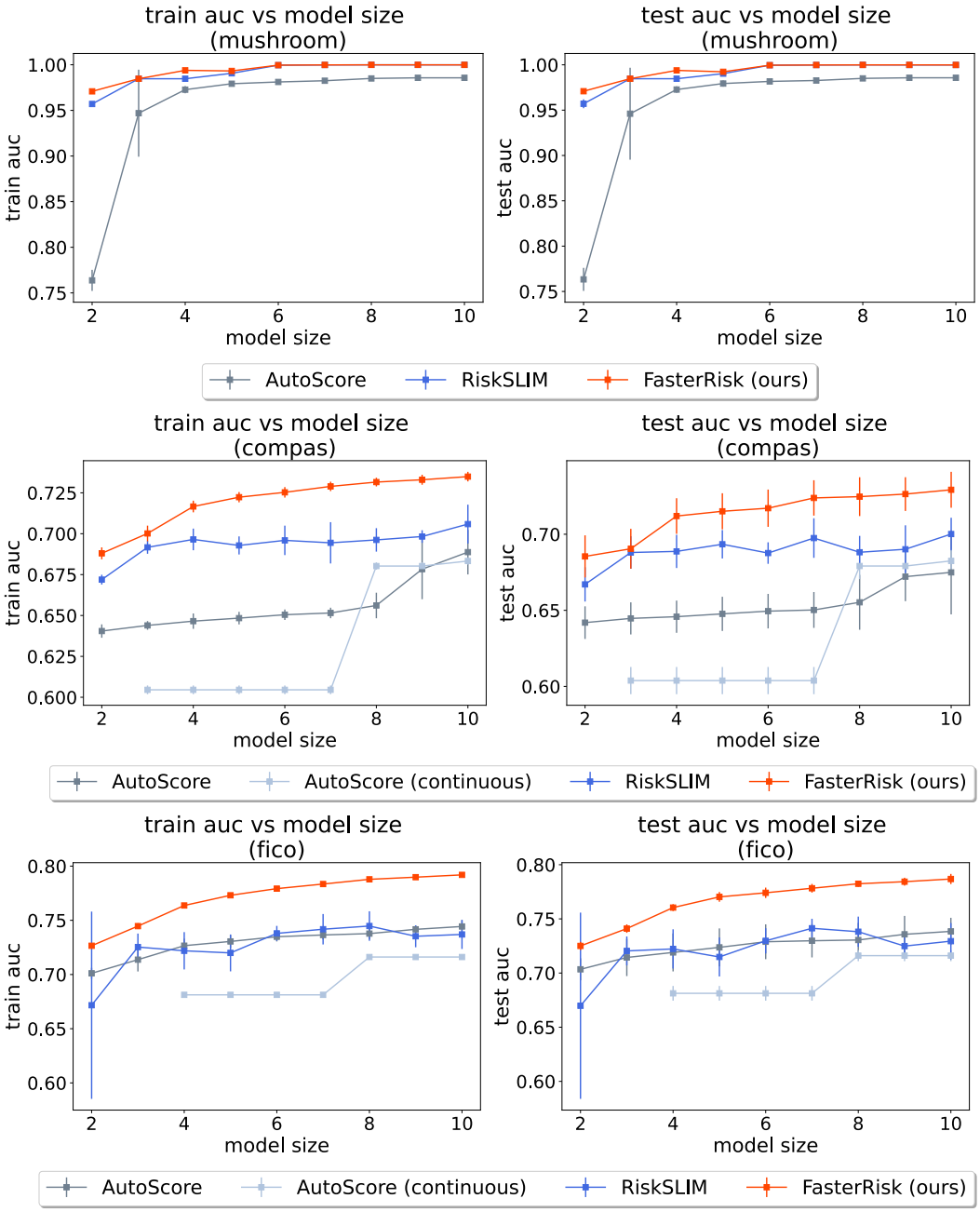

Figure 40: Comparison with the new baseline on the mushroom, Compas, and FICO datasets. The AutoScore (continuous) baseline is another method where AutoScore is applied to the original continuous features instead of the binary features as detailed in Appendix D.1. Not every model size can be obtained by the AutoScore (continuous) method. The left column is training AUC (higher is better), and the right column is test AUC (higher is better).

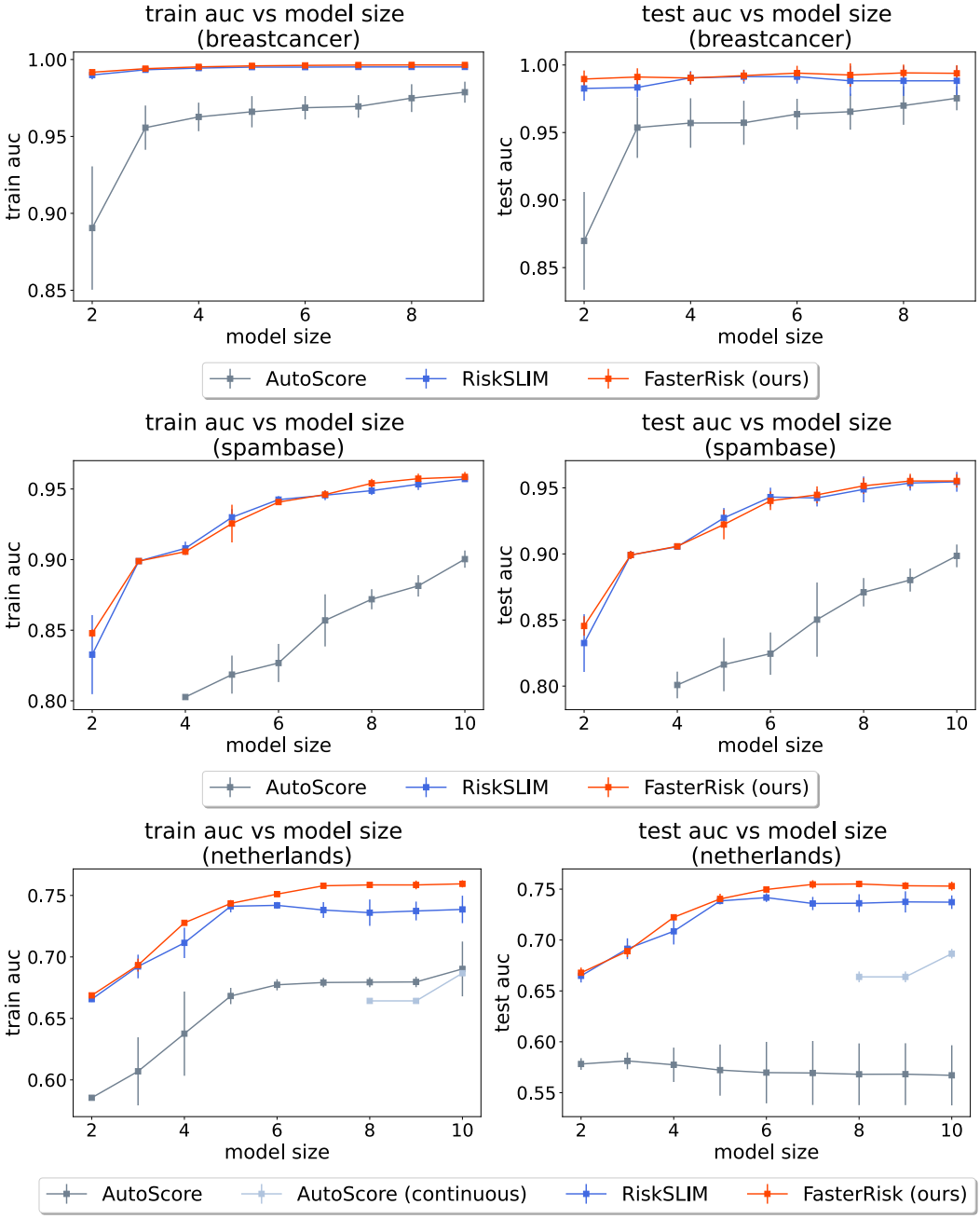

Figure 41: Comparison with the new baseline on the breastcancer, spambase, and Netherlands datasets. The AutoScore (continuous) baseline is another method where AutoScore is applied to the original continuous features instead of the binary features as detailed in Appendix D.1. Not every model size can be obtained by the AutoScore (continuous) method. The left column is training AUC (higher is better), and the right column is test AUC (higher is better).

F Additional Risk Score Models

We provide additional risk score models for the readers to inspect.

Appendix F.1 shows risk scores with different model sizes on different datasets.

Appendix F.2 shows different risk scores with the same size from the diverse pool of solutions.

Specifically, Appendix F.2.1 shows different risk scores on the bank dataset (financial application), Appendix F.2.2 shows different risk scores on the mammo dataset (medical application), and Appendix F.2.3 shows different risk scores on the Netherlands dataset (criminal justice application).

F.1 Risk Score Models with Different Sizes

For model size $= 3$, please see Tables 3-11.

For model size $= 5$, please see Tables 12-20.

For model size $= 7$, please see Tables 21-29.

We also include a large model with size $= 10$ on the FICO dataset, please see Table 30.

1.	no high school diploma	-4 points		...
2.	high school diploma only	-2 points	+	...
3.	married	4 points	+	...
		SCORE	=	

SCORE	-4	-2	0	2	4
RISK	1.2%	4.1%	13.1%	34.7%	65.3%

Table 3: FasterRisk model for the adult dataset, predicting salary$>$ 50K.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Employment Indicator $<$ 5100	4 points	+	...
		SCORE	=	

SCORE	-2	0	2	6	8
RISK	2.8%	6.5%	14.5%	50.0%	70.8%

Table 4: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call.

1.	Irregular Shape	4 points		...
2.	Circumscribed Margin	-5 points	+	...
3.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-5	-2	-1	2
RISK	8.2%	20.1%	26.2%	50.0%

SCORE	3	4	7
RISK	58.5%	66.6%	84.9%

Table 5: FasterRisk model for the mammo dataset, predicting malignancy of a breast lesion.

1.	odor=almond	-5 points		...
2.	odor=anise	-5 points	+	...
3.	odor=none	-5 points	+	...
		SCORE	=	

SCORE	-5	0
RISK	10.8%	96.0%

Table 6: FasterRisk model for the mushroom dataset, predicting whether a mushroom is poisonous.

1.	prior_counts ≤ 2	-4 points		...
2.	prior_counts ≤ 7	-4 points	+	...
3.	age ≤ 31	4 points	+	...
		SCORE	=	

SCORE	-8	-4	0	4
RISK	23.6%	44.1%	67.0%	83.9%

Table 7: FasterRisk model for the COMPAS dataset, predicting whether individuals are arrested within two years of release.

1.	MSinceMostRecentInqexcl7days≤ 0	3 points		...
2.	ExternalRiskEstimate≤ 70	5 points	+	...
3.	ExternalRiskEstimate≤ 79	5 points	+	...
		SCORE	=	

SCORE	0	3	5	8	≥ 10
RISK	13.7%	24.0%	33.4%	50.0%	$\geq$ 61.3%

Table 8: FasterRisk model for the FICO dataset, predicting whether an individual will default on a loan.

1.	Clump Thickness	$\times$3 points		...
2.	Uniformity of Cell Size	$\times$5 points	+	...
3.	Bare Nuclei	$\times$3 points	+	...
		SCORE	=	

SCORE	≤ 33	36	39	42	45
RISK	$\leq$ 3.3%	6.1%	10.8%	18.6%	30.1

SCORE	48	51	54	57	≥ 60
RISK	67.0%	77.6%	85.5%	91.0	$\geq$ 94.5%

Table 9: FasterRisk model for the breastcancer dataset, predicting whether there is breast cancer using a biopsy.

1.	WordFrequency_Remove	$\times$5 points		...
2.	WordFrequency_HP	$\times$-2 points	+	...
3.	CharacterFrequency_$	$\times$5 points	+	...
		SCORE	=	

SCORE	$\leq$ -4	-3	-2	-1	0
RISK	$\leq$0.4%	1.3%	3.7%	10.2%	25.2%

SCORE	1	2	3	4	≥ 5
RISK	50.0%	74.8%	89.8%	96.3%	$\geq$ 98.7%

Table 10: FasterRisk model for the spambase dataset, predicting if an e-mail is spam.

1.	previous case ≤ 20	-5 points		...
2.	previous case ≤ 10 or previous case ≥ 21	-4 points	+	...
3.	# of previous penal cases≤ 3	-2 points	+	...
		SCORE	=	

SCORE	≤ -9	-7	-6	0
RISK	$\leq$ 50%	74.6%	83.4%	99.2%

Table 11: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years.

1.	no high school diploma	-4 points		...
2.	high school diploma only	-2 points	+	...
3.	age 22 to 29	-2 points	+	...
4.	any capital gains	3 points	+	...
5.	married	4 points	+	...
		SCORE	=	

SCORE	<-4	-3	-2	-1	0
RISK	<1.3%	2.4%	4.4%	7.8%	13.6%

SCORE	1	2	3	4	7
RISK	22.5%	35.0%	50.5%	65.0%	92.2%

Table 12: FasterRisk model for the adult dataset, predicting salary> 50K. This table has already been shown in the main paper.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Previous Marketing Campaign Failed	-1 points	+	...
4.	Employment Indicator > 5100	-5 points	+	...
5.	3 Month Euribor Rate ≥ 100	-2 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 11.2%	15.1%	20.1%	26.2%	33.4%
SCORE	0	1	2	3	4
RISK	41.5%	50.0%	58.5%	66.6%	73.8%

Table 13: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call.

1.	Oval Shape	-2 points		...
2.	Irregular Shape	4 points	+	...
3.	Circumscribed Margin	-5 points	+	...
4.	Spiculated Margin	2 points	+	...
5.	Age ≥ 60	3 points	+	
		SCORE	=	

SCORE	-7	-5	-4	-3	-2	-1
RISK	6.0%	10.6%	13.8%	17.9%	22.8%	28.6%

SCORE	0	1	2	3	4	≥ 5
RISK	35.2%	42.4%	50.0%	57.6%	64.8%	71.4%

Table 14: FasterRisk model for the mammo dataset, predicting malignancy of a breast lesion. This table has already been shown in the main paper.

1.	odor=almond	-5 points		...
2.	odor=anise	-5 points	+	...
3.	odor=none	-5 points	+	...
4.	odor=foul	5 points	+	...
5.	gill size=broad	-3 points	+	...
		SCORE	=	

SCORE	-8	-5	-3	$\geq$2
RISK	1.62%	26.4%	73.6%	>99.8%

Table 15: FasterRisk model for the mushroom dataset, predicting whether a mushroom is poisonous. This table has already been shown in the main paper.

1.	prior_counts ≤ 7	-5 points		...
2.	prior_counts ≤ 2	-5 points	+	...
3.	prior_counts ≤ 0	-3 points	+	...
4.	age ≤ 33	4 points	+	...
5.	age ≤ 23	5 points	+	...
		SCORE	=	

SCORE	$\leq$ -10	-9	-6	-5	-4
RISK	$\leq$25.9%	29.4%	41.3%	45.6%	50.0%

SCORE	-2	-1	3	4	9
RISK	58.7%	62.8%	77.3%	80.2%	$\geq$ 90.7%

Table 16: FasterRisk model for the COMPAS dataset, predicting whether individuals are arrested within two years of release.

1.	MSinceMostRecentInqexcl7days ≤ -8	-4 points		...
2.	MSinceMostRecentInqexcl7days ≤ 0	2 points	+	...
3.	NumSatisfactoryTrades ≤ 12	2 points	+	...
4.	ExternalRiskEstimate ≤ 70	3 points	+	...
5.	ExternalRiskEstimate ≤ 79	3 points	+	...
		SCORE	=	

SCORE	-2	0	1	2	3
RISK	6.7%	13.2%	18.2%	24.4%	32.0%

SCORE	4	5	6	8	10
RISK	40.7%	50.0%	59.3%	75.5%	86.8%

Table 17: FasterRisk model for the FICO dataset, predicting whether an individual will default on a loan. -8 means a missing value on the FICO dataset.

1.	Clump Thickness	$\times$5 points		...
2.	Uniformity of Cell Size	$\times$4 points	+	...
3.	Marginal Adhesion	$\times$3 points	+	...
4.	Bare Nuclei	$\times$4 points	+	...
5.	Normal Nucleoli	$\times$3 points	+	...
		SCORE	=	

SCORE	$\leq$ 55	60	65	70	75
RISK	$\leq$ 8.6%	14.6%	23.5%	35.7%	50.0

SCORE	80	85	90	95	$\geq$ 100
RISK	64.3%	76.5%	85.4%	91.4	$\geq$ 95.0%

Table 18: FasterRisk model for the breastcancer dataset, predicting whether there is breast cancer using a biopsy.

1.	WordFrequency_Remove	$\times$5 points		...
2.	WordFrequency_Free	$\times$2 points		...
3.	WordFrequency_0	$\times$5 points	+	...
4.	WordFrequency_HP	$\times$-2 points	+	...
5.	WordFrequency_George	$\times$-2 points	+	...
		SCORE	=	

SCORE	$\leq$ -4	-3	-2	-1	0
RISK	$\leq$0.6%	1.6%	4.4%	11.4%	26.4%

SCORE	1	2	3	4	$\geq$ 5
RISK	50.0%	73.6%	88.6%	95.6%	$\geq$ 98.4%

Table 19: FasterRisk model for the spambase dataset, predicting if an e-mail is spam.

1.	previous case ≤ 20	-5 points		...
2.	previous case ≤ 10 or previous case ≥ 21	-3 points	+	...
3.	# of previous penal cases ≤ 2	-2 points	+	...
4.	age in years ≤ 38.06	1 points	+	...
5.	age at first penal case ≤ 22.63	1 points	+	...
		SCORE	=	

SCORE	$\leq$-9	-8	-7	-6	-5	-4
RISK	$\leq$ 23.8%	35.8%	50.0%	64.2%	76.2%	85.1%

SCORE	-3	-2	-1	0	1	2
RISK	91.1%	94.8%	97.0%	98.3%	99.1%	99.5%

Table 20: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years.

1.	Age 22 to 29	-2 points		...
2.	High School Diploma Only	-2 points	+	...
3.	No High school Diploma	-4 points		...
4.	Married	4 points	+	...
5.	Work Hours Per Week < 50	-2 points	+	...
6.	Any Capital Gains	3 points	+	...
7.	Any Capital Loss	2 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$0.8%	1.4%	2.6%	4.6%	8.1%
SCORE	0	2	3	4	7
RISK	14.0%	35.3%	50.0%	64.7%	91.9%

Table 21: FasterRisk model for the adult dataset, predicting salary> 50K.

1.	Blue Collar Job	-1 points		...
2.	Call in Second Quarter	-2 points	+	...
3.	Previous Call Was Successful	3 points	+	...
4.	Previous Marketing Campaign Failed	-1 points	+	...
5.	Employment Indicator $>$ 5100	-5 points	+	...
6.	Consumer Price Index $\geq$ 93.5	1 points	+	...
7.	3 Month Euribor Rate $\geq$ 100	-1 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 7.9%	11.5%	16.3%	22.7%	30.6%
SCORE	0	1	2	3	4
RISK	39.9%	50.0%	60.1%	69.4%	77.3%

Table 22: FasterRisk model for the bank dataset, predicting if a person opens bank account after marketing call.

1.	Lobular Shape	2 points		...
2.	Irregular Shape	5 points	+	...
3.	Circumscribed Margin	-4 points	+	...
4.	Obscured Margin	-1 points	+	...
5.	Spiculated Margin	1 points	+	...
6.	Age $<$ 30	-5 points	+	...
7.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	$\leq$-1	0	1	2	3
RISK	19.8%	25.9%	33.2%	41.3%	50.0%
SCORE	4	5	6	8	9
RISK	58.7%	66.8%	74.1%	85.2%	89.1%

Table 23: FasterRisk model for the mammo dataset, predicting malignancy of a breast lesion.

1.	odor=anise	-5 points		...
2.	odor=none	-5 points	+	...
3.	odor=foul	5 points	+	...
4.	gill size=narrow	4 points	+	...
5.	stalk surface above ring=grooves	2 points	+	...
6.	spore print color=green	5 points	+	...
		SCORE	=	

SCORE	-5	0	2	4	$\geq$ 5
RISK	0.5%	50.0%	89.2%	98.6%	99.5%

Table 24: FasterRisk model for the mushroom dataset, predicting whether a mushroom is poisonous.

1.	prior_counts ≤ 7	-3 points		...
2.	prior_counts ≤ 2	-3 points	+	...
3.	prior_counts ≤ 0	-2 points	+	...
4.	age ≤ 52	2 points	+	...
5.	age ≤ 33	2 points	+	...
6.	age ≤ 23	2 points	+	...
7.	age ≤ 20	4 points	+	...
		SCORE	=	

SCORE	-8	-6	-4	-3	-2	-1	0
RISK	11.3%	18.7%	29.3%	35.7%	42.7%	50.0%	57.3%

SCORE	1	2	3	4	6	7	10
RISK	64.3%	70.7%	76.4%	81.3%	88.7%	91.3%	96.2%

Table 25: FasterRisk model for the COMPAS dataset, predicting whether individuals are arrested within two years of release.

1.	MSinceMostRecentInqexcl7days ≤ -8	-4 points		...
2.	MSinceMostRecentInqexcl7days ≤ 0	2 points	+	...
3.	NetFractionRevolvingBurden ≤ 37	-2 points	+	...
4.	ExternalRiskEstimate ≤ 70	2 points	+	...
5.	ExternalRiskEstimate ≤ 78	2 points	+	...
6.	AverageMInFile ≤ 60	2 points	+	...
7.	PercentTradesNeverDelq ≤ 85	2 points	+	...
		SCORE	=	

SCORE	-4	-2	0	2	4	6	8	10
RISK	8.0%	14.9%	26.0%	41.4%	58.6%	74.0%	85.1%	92.0%

Table 26: FasterRisk model for the FICO dataset, predicting whether an individual will default on a loan. -8 means a missing value on the FICO dataset.

1.	Clump Thickness	$\times 4$ points		...
2.	Uniformity of Cell Shape	$\times 3$ points	+	...
3.	Marginal Adhesion	$\times 3$ points	+	...
4.	Bare Nuclei	$\times 3$ points	+	...
5.	Bland Chromatin	$\times 3$ points	+	...
6.	Normal Nucleoli	$\times 2$ points	+	...
7.	Mitoses	$\times 4$ points	+	...
		SCORE	=	

SCORE	≤ 55	60	65	70	75
RISK	$\leq 5.1\%$	9.3%	16.2%	26.6%	40.6

SCORE	80	85	90	95	≥ 100
RISK	56.3%	70.8%	82.1%	89.6	$\geq 94.2\%$

Table 27: FasterRisk model for the breastcancer dataset, predicting whether there is breast cancer using a biopsy.

1.	WordFrequency_Remove	$\times$4 points		...
2.	WordFrequency_Free	$\times$2 points		...
3.	WordFrequency_Business	$\times$1 points	+	...
4.	WordFrequency_0	$\times$4 points	+	...
5.	WordFrequency_HP	$\times$-2 points	+	...
6.	WordFrequency_George	$\times$-2 points	+	...
7.	CharacterFrequency_$	$\times$5 points	+	...
		SCORE	=	

SCORE	$\leq$ -4	-3	-2	-1	0
RISK	$\leq$0.4%	1.3%	3.7%	10.2%	25.2%

SCORE	1	2	3	4	$\geq$ 5
RISK	50.0%	74.8%	89.8%	96.3%	$\geq$ 98.7%

Table 28: FasterRisk model for the spambase dataset, predicting if an e-mail is spam.

1.	previous case ≤ 20	-5 points		...
2.	previous case ≤ 10 or previous case ≥ 21	-4 points	+	...
3.	# of previous penal cases ≤ 1	-1 points	+	...
4.	# of previous penal cases ≤ 3	-1 points	+	...
5.	# of previous penal cases ≤ 5	-1 points	+	...
6.	age in years ≤ 21.80	1 points	+	...
7.	age in years ≤ 38.05	1 points	+	...
		SCORE	=	

SCORE	$\leq$-10	-9	-8	-7	-6	-5
RISK	$\leq$ 33.1%	50.0%	66.9%	80.3%	89.2%	94.3%

SCORE	-4	-3	-2	-1	0	$\geq$ 1
RISK	97.1%	98.6%	99.3%	99.6%	99.8%	99.9%

Table 29: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years.

1.	ExternalRiskEstimate$\leq$63	1 points		...
2.	ExternalRiskEstimate$\leq$70	2 points	+	...
3.	ExternalRiskEstimate$\leq$79	2 points	+	...
4.	AverageMInFile$\leq$59	2 points	+	...
5.	NumSatisfactoryTrades$\leq$13	2 points	+	...
6.	PercentTradesNeverDelq$\leq$95	1 points	+	...
7.	PercentInstallTrades$\leq$46	-1 points	+	...
8.	MSinceMostRecentInqexcl7days$\leq$-8	-5 points	+	...
9.	MSinceMostRecentInqexcl7days$\leq$0	2 points	+	...
10.	NetFractionRevolvingBurden$\leq$37	-2 points	+	...
		SCORE	=	

SCORE	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2
RISK	2.7%	3.7%	5.0%	6.8%	9.2%	12.4%	16.4%	21.3%	27.3%	34.2%	41.9%

SCORE	3	4	5	6	7	8	9	10	11	12
RISK	50.0%	58.1%	65.8%	72.7%	78.7%	83.6%	87.6%	90.8%	93.2%	95.0%

Table 30: FasterRisk model for the FICO dataset, predicting whether an individual will default on a loan. -8 means a missing value on the FICO dataset.

F.2 Risk Score Models from the Pool of Solutions

F.2.1 Examples from the Pool of Solutions (Bank Dataset)

The extra risk score examples from the pool of solutions are shown in Tables 31-42. All models were from the pool of the third fold on the bank dataset, and we show the top 12 models, provided in ascending order of the logistic loss on the training set (the model with the smallest logistic loss comes first).

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Previous Marketing Campaign Failed	-1 points	+	...
4.	Employment Indicator $>$ 5100	-5 points	+	...
5.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 11.2%	15.1%	20.1%	26.2%	33.4%
SCORE	0	1	2	3	4
RISK	41.5%	50.0%	58.5%	66.6%	73.8%

Table 31: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9352.39. The AUCs on the training and test sets are 0.779 and 0.770, respectively.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Previous Marketing Campaign Failed	-1 points	+	...
4.	Employment Variation Rate $<$ -1	5 points	+	...
5.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
		SCORE	=	

SCORE	$\leq$0	1	2	3	4
RISK	$\leq$ 11.2%	15.1%	20.1%	26.2%	33.4%
SCORE	5	6	7	8	9
RISK	41.5%	50.0%	58.5%	66.6%	73.8%

Table 32: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9352.39. The AUCs on the training and test sets are 0.779 and 0.770, respectively.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Previous Marketing Campaign Failed	-1 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-5 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 11.2%	15.2%	20.1%	26.3%	33.4%
SCORE	0	1	2	3	4
RISK	41.5%	50.0%	58.5%	66.6%	73.7%

Table 33: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9352.86. The AUCs on the training and test sets are 0.779 and 0.769, respectively.

1.	Call in Second Quarter	-2 points		...
2.	Previous Marketing Campaign Failed	-1 points	+	...
3.	Previous Marketing Campaign Succeeded	4 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-5 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 11.3%	15.3%	20.2%	26.3%	33.5%
SCORE	0	1	2	3	4
RISK	41.5%	50.0%	58.5%	66.5%	73.6%

Table 34: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9363.40. The AUCs on the training and test sets are 0.779 and 0.769, respectively. Note that some customers do not have previous marketing campaigns, so for these customers, neither of conditions 2 nor 3 are satisfied.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	4 points	+	...
3.	Consumer Price Index $>$ 93.5	1 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-1 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-5 points	+	...
		SCORE	=	

SCORE	$\leq$-4	-3	-2	-1	0
RISK	$\leq$ 9.6%	13.4%	18.4%	24.6%	32.2%
SCORE	1	2	3	4	5
RISK	40.8%	50.0%	59.2%	67.8%	75.4%

Table 35: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9365.51. The AUCs on the training and test sets are 0.778 and 0.769, respectively.

1.	Call in First Quarter	2 points		...
2.	Call in Second Quarter	-1 points		...
3.	Previous Call Was Successful	3 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-3 points	+	...
		SCORE	=	

SCORE	$\leq$-4	-3	-2	-1	0
RISK	$\leq$ 8.8%	13.4%	19.8%	28.2%	38.5%
SCORE	1	2	3	4	5
RISK	50.0%	61.5%	71.8%	80.2%	86.6%

Table 36: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9365.57. The AUCs on the training and test sets are 0.776 and 0.766, respectively.

1.	Call in Second Quarter	-2 points		...
2.	Previous Call Was Successful	3 points	+	...
3.	Previous Marketing Campaign Failed	-1 points	+	...
4.	Consumer Price Index $\geq$ 93.5	1 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-4 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 3.0%	5.2%	9.0%	15.0%	23.9%
SCORE	0	1	2	3	4
RISK	35.9%	50.0%	64.1%	76.1%	85.0%

Table 37: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9367.20. The AUCs on the training and test sets are 0.781 and 0.772, respectively.

1.	Call in Second Quarter	-1 points		...
2.	Previous Call Was Successful	5 points	+	...
3.	Calls Before Campaign Succeeded	-1 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-4 points	+	...
		SCORE	=	

SCORE	$\leq$-4	-3	-2	-1	0
RISK	$\leq$ 11.4%	16.3%	22.6%	30.6%	39.9%
SCORE	1	2	3	4	5
RISK	50.0%	60.1%	69.4%	77.4%	83.7%

Table 38: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9367.93. The AUCs on the training and test sets are 0.780 and 0.769, respectively.

1.	Call in Second Quarter	-1 points		...
2.	Any Prior Calls Before Campaign	4 points	+	...
3.	Previous Marketing Campaign Failed	-5 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-4 points	+	...
		SCORE	=	

SCORE	$\leq$-5	-4	-3	-2	-1
RISK	$\leq$ 7.8%	11.4%	16.2%	22.6%	30.5%
SCORE	0	1	2	3	4
RISK	39.9%	50.0%	60.1%	69.5%	77.4%

Table 39: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9371.75. The AUCs on the training and test sets are 0.779 and 0.769, respectively.

1.	Called via Landline Phone	-2 points		...
2.	Previous Call Was Successful	5 points	+	...
3.	Previous Marketing Campaign Failed	-2 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-4 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-4 points	+	...
		SCORE	=	

SCORE	$\leq$-6	-5	-4	-3	-2
RISK	$\leq$ 10.9%	14.2%	18.3%	23.2%	28.9%
SCORE	-1	0	1	3	5
RISK	35.4%	42.6%	50.0%	64.6%	76.8%

Table 40: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9376.52. The AUCs on the training and test sets are 0.776 and 0.765, respectively.

1.	Job Is Retired	1 points		...
2.	Call in Second Quarter	-2 points	+	...
3.	Previous Call Was Successful	5 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-5 points	+	...
		SCORE	=	

SCORE	$\leq$-3	-2	-1	0	1
RISK	$\leq$ 17.2%	22.2%	28.1%	34.8%	42.2%
SCORE	2	3	4	5	6
RISK	50.0%	57.8%	65.2%	72.0%	77.8%

Table 41: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9378.18. The AUCs on the training and test sets are 0.777 and 0.766, respectively.

1.	Age $\geq$ 60	1 points		...
2.	Call in Second Quarter	-2 points	+	...
3.	Previous Call Was Successful	5 points	+	...
4.	3 Month Euribor Rate $\geq$ 100	-2 points	+	...
5.	3 Month Euribor Rate $\geq$ 200	-5 points	+	...
		SCORE	=	

SCORE	$\leq$-3	-2	-1	0	1
RISK	$\leq$ 17.3%	22.2%	28.1%	34.8%	42.2%
SCORE	2	3	4	5	6
RISK	50.0%	57.8%	65.2%	71.9%	77.8%

Table 42: FasterRisk model for the bank dataset, predicting if a person opens a bank account after a marketing call. The logistic loss on the training set is 9378.68. The AUCs on the training and test sets are 0.777 and 0.767, respectively.

F.2.2 Examples from the Pool of Solutions (Mammo Dataset)

The extra risk score examples from the pool of solutions are shown in Tables 43-54. All models were from the pool of the third fold on the mammo dataset, and we show the top 12 models, provided in ascending order of the logistic loss on the training set (the model with the smallest logistic loss comes first).

1.	Oval Shape	-2 points		...
2.	Irregular Shape	4 points	+	...
3.	Circumscribed Margin	-5 points	+	...
4.	Spiculated Margin	2 points	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-7	-5	-4	-3	-2	-1
RISK	6.0%	10.6%	13.8%	17.9%	22.8%	28.6%

SCORE	0	1	2	3	4	$\geq$ 5
RISK	35.2%	42.4%	50.0%	57.6%	64.8%	71.4%

Table 43: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 357.77. The AUCs on the training and test sets are 0.854 and 0.853, respectively.

1.	Lobular Shape	1 point		...
2.	Irregular Shape	3 points	+	...
3.	Circumscribed Margin	-3 points	+	...
4.	Spiculated Margin	1 point	+	...
5.	Age $\geq$ 60	2 points	+	...
		SCORE	=	

SCORE	-3	-2	-1	0	1
RISK	7.5%	11.8%	18.1%	26.8%	37.7%
SCORE	2	3	4	5	6
RISK	50.0%	62.3%	73.2%	81.9%	88.2%

Table 44: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 357.86. The AUCs on the training and test sets are 0.854 and 0.857, respectively.

1.	Lobular Shape	2 points		...
2.	Irregular Shape	5 points	+	...
3.	Circumscribed Margin	-4 points	+	...
4.	Age $<$ 30	-5 points	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-9	-7	-6	-5	-4	-3	-2	-1	0
RISK	1.3%	2.6%	3.7%	5.2%	7.3%	10.1%	14.0%	18.9%	25.1%
SCORE	1	2	3	4	5	6	7	8	10
RISK	32.6%	41.0%	50.0%	59.0%	67.4%	74.9%	81.1%	86.0%	92.7%

Table 45: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.24. The AUCs on the training and test sets are 0.852 and 0.854, respectively.

1.	Lobular Shape	2 points	...
2.	Irregular Shape	5 points	+ ...
3.	Circumscribed Margin	-4 points	+ ...
4.	Age $\geq$ 30	5 points	+ ...
5.	Age $\geq$ 60	3 points	+ ...
		SCORE	=

SCORE	-4	-2	-1	0	1	2	3	4	5
RISK	1.3%	2.6%	3.7%	5.2%	7.3%	10.1%	14.0%	18.9%	25.1%
SCORE	6	7	8	9	10	11	12	13	15
RISK	32.6%	41.0%	50.0%	59.0%	67.4%	74.9%	81.1%	86.0%	92.7%

Table 46: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.24. The AUCs on the training and test sets are 0.852 and 0.854, respectively.

1.	Irregular Shape	2 points	...
2.	Circumscribed Margin	-2 points	+ ...
3.	Spiculated Margin	1 point	+ ...
4.	Age $\geq$ 30	2 points	+ ...
5.	Age $\geq$ 60	1 point	+ ...
		SCORE	=

SCORE	-2	-1	0	1	2
RISK	2.3%	4.7%	9.5%	18.2%	32.0%

SCORE	3	4	5	6
RISK	50.0%	68.0%	81.8%	90.5%

Table 47: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.59. The AUCs on the training and test sets are 0.852 and 0.857, respectively.

1.	Irregular Shape	2 points	...
2.	Circumscribed Margin	-2 points	+ ...
3.	Spiculated Margin	1 point	+ ...
4.	Age $<$ 30	-2 points	+ ...
5.	Age $\geq$ 60	1 point	+ ...
		SCORE	=

SCORE	-4	-3	-2	-1	0
RISK	2.3%	4.7%	9.5%	18.2%	32.0%

SCORE	1	2	3	4
RISK	50.0%	68.0%	81.8%	90.5%

Table 48: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.59. The AUCs on the training and test sets are 0.852 and 0.857, respectively.

1.	Lobular Shape	2 points		...
2.	Irregular Shape	5 points	+	...
3.	Circumscribed Margin	-4 points	+	...
4.	Obscure Margin	-1 point	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-5	-4	-3	-2	-1	0	1	2
RISK	5.3%	7.4%	10.3%	14.1%	19.1%	25.3%	32.7%	41.1%
SCORE	3	4	5	6	7	8	9	10
RISK	50.0%	58.9%	67.3%	74.7%	80.9%	85.9%	89.7%	92.6%

Table 49: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.71. The AUCs on the training and test sets are 0.852 and 0.857, respectively.

1.	Irregular Shape	5 points		...
2.	Circumscribed Margin	-5 points	+	...
3.	Microlobulated Margin	2 points	+	...
4.	Spiculated Margin	2 points	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-5	-3	-2	-1	0	2	3
RISK	8.6%	14.6%	18.6%	23.5%	29.2%	42.7%	50.0%
SCORE	4	5	7	8	9	10	12
RISK	57.3%	64.3%	76.5%	81.4%	85.4%	88.7%	93.4%

Table 50: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 358.98. The AUCs on the training and test sets are 0.852 and 0.852, respectively.

1.	Irregular Shape	4 points		...
2.	Circumscribed Margin	-5 points	+	...
3.	SpiculatedMargin	2 points	+	...
4.	Age $\geq$ 45	1 point	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-5	-4	-3	-2	-1	0	1	2
RISK	7.3%	9.7%	12.9%	16.9%	21.9%	27.8%	34.6%	42.1%
SCORE	3	4	5	6	7	8	9	10
RISK	50.0%	57.9%	65.4%	72.2%	78.1%	83.1%	87.1%	90.3%

Table 51: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 359.10. The AUCs on the training and test sets are 0.855 and 0.859, respectively.

1.	Irregular Shape	4 points		...
2.	Circumscribed Margin	-5 points	+	...
3.	Obscure Margin	-1 points	+	...
4.	Spiculated Margin	2 points	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-6	-5	-4	-3	-2	-1	0	1
RISK	6.8%	9.2%	12.3%	16.3%	21.3%	27.3%	34.2%	41.9%
SCORE	2	3	4	5	6	7	8	9
RISK	50.0%	58.1%	65.8%	72.7%	78.7%	83.7%	87.7%	90.8%

Table 52: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 359.34. The AUCs on the training and test sets are 0.852 and 0.862, respectively.

1.	Oval Shape	-1 point		...
2.	Lobular Shape	1 point	+	...
3.	Irregular Shape	4 points	+	...
4.	Circumscribed Margin	-4 points	+	...
5.	Age $\geq$ 60	3 points	+	...
		SCORE	=	

SCORE	-5	-4	-3	-2	-1	0	1
RISK	7.0%	9.8%	13.6%	18.5%	24.8%	32.3%	40.8%
SCORE	2	3	4	5	6	7	8
RISK	50.0%	59.2%	67.7%	75.2%	81.5%	86.4%	90.2%

Table 53: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 359.53. The AUCs on the training and test sets are 0.850 and 0.849, respectively.

1.	Lobular Shape	1 point		...
2.	Irregular Shape	4 points	+	...
3.	Circumscribed Margin	-3 points	+	...
4.	Age $\geq$ 45	1 point	+	...
5.	Age $\geq$ 60	2 points	+	...
		SCORE	=	

SCORE	-3	-2	-1	0	1	2
RISK	6.3%	9.5%	14.1%	20.5%	28.9%	38.9%
SCORE	3	4	5	6	7	8
RISK	50.0%	61.1%	71.1%	79.5%	85.9%	90.5%

Table 54: FasterRisk model for the mammo dataset, predicting the risk of malignancy of a breast lesion. The logistic loss on the training set is 359.53. The AUCs on the training and test sets are 0.852 and 0.850, respectively.

F.2.3 Examples from the Pool of Solutions (Netherlands Dataset)

The extra risk score examples from the pool of solutions are shown in Tables 55-66. All models were from the pool of the third fold on the Netherlands dataset, and we show the top 12 models, provided in ascending order of the logistic loss on the training set (the model with the smallest logistic loss comes first).

1.	# of previous penal cases ≤ 2	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.633	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	14.9%	23.8%	35.8%	50.0%	64.2%	76.2%	85.1%

SCORE	-3	-2	-1	0	1	2
RISK	91.1%	94.8%	97.0%	98.3%	99.1%	99.5%

Table 55: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9226.84. The AUCs on the training and test sets are 0.743 and 0.742, respectively.

1.	# of previous penal cases ≤ 1	-1 point		...
2.	# of previous penal cases ≤ 3	-1 point	+	...
3.	age in years ≤ 38.052	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-3 points	+	...
		SCORE	=	

SCORE	-8	-7	-6	-5	-4
RISK	12.4%	27.4%	50.0%	72.6%	87.6%

SCORE	-3	-2	-1	0	1
RISK	94.9%	98.0%	99.2%	99.7%	99.9%

Table 56: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9232.51. The AUCs on the training and test sets are 0.744 and 0.739, respectively.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 23.265	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	24.9%	36.6%	50.0%	63.4%	75.1%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 57: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9250.94. The AUCs on the training and test sets are 0.739 and 0.739, respectively.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.989	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	25.0%	36.6%	50.0%	63.4%	75.0%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.0%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 58: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9250.95. The AUCs on the training and test sets are 0.738 and 0.739, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 23.283	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.0%	24.9%	36.6%	50.0%	63.4%	75.1%	84.0%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 59: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9251.14. The AUCs on the training and test sets are 0.739 and 0.739, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.934	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	25.0%	36.6%	50.0%	63.4%	75.0%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.0%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 60: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9251.39. The AUCs on the training and test sets are 0.739 and 0.740, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.907	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	24.9%	36.6%	50.0%	63.4%	75.1%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.0%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 61: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9251.53. The AUCs on the training and test sets are 0.739 and 0.740, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 23.328	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.0%	24.9%	36.5%	50.0%	63.5%	75.1%	84.0%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 62: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9252.07. The AUCs on the training and test sets are 0.738 and 0.739, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.965	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	24.9%	36.6%	50.0%	63.4%	75.1%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 63: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9252.13. The AUCs on the training and test sets are 0.738 and 0.740, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.830	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	24.9%	36.6%	50.0%	63.4%	75.1%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 64: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9252.19. The AUCs on the training and test sets are 0.739 and 0.740, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.870	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.1%	24.9%	36.6%	50.0%	63.4%	75.1%	83.9%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 65: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9252.25. The AUCs on the training and test sets are 0.739 and 0.740, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

1.	# of previous penal cases ≤ 3	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 23.233	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	16.0%	24.9%	36.5%	50.0%	63.5%	75.1%	84.0%

SCORE	-3	-2	-1	0	1	2
RISK	90.1%	94.0%	96.5%	97.9%	98.8%	99.3%

Table 66: FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. The logistic loss on the training set is 9252.27. The AUCs on the training and test sets are 0.738 and 0.739, respectively. Note that this risk score is slightly different from that of Table 57 in Condition 3.

G Model Reduction

G.1 Reducing Models to Relatively Prime Coefficients

If the coefficients of a model are not relatively prime, one can divide all the coefficients by any common prime factors without changing any of the predicted risks. Table 67(left), copied from Table 6, is reduced in this way to produce Table 67(right). Table 68(left), copied from Table 7, is reduced in this way to produce Table 68(right).

1.	odor=almond	-5 points		...
2.	odor=anise	-5 points	+	...
3.	odor=none	-5 points	+	...
		SCORE	=	

SCORE	-5	0
RISK	10.8%	96.0%

1.	odor=almond	-1 points		...
2.	odor=anise	-1 points	+	...
3.	odor=none	-1 points	+	...
		SCORE	=	

SCORE	-1	0
RISK	10.8%	96.0%

Table 67: *Left:* FasterRisk model for the Mushroom dataset, predicting whether a mushroom is poisonous. Copy of Table 6. *Right:* Reduction to have relatively prime coefficients.

1.	prior_counts ≤ 2	-4 points		...
2.	prior_counts ≤ 7	-4 points	+	...
3.	age ≤ 31	4 points	+	...
		SCORE	=	

SCORE	-8	-4	0	4
RISK	23.6%	44.1%	67.0%	83.9%

1.	prior_counts ≤ 2	-1 points		...
2.	prior_counts ≤ 7	-1 points	+	...
3.	age ≤ 31	1 points	+	...
		SCORE	=	

SCORE	-2	-1	0	1
RISK	23.6%	44.1%	67.0%	83.9%

Table 68: *Left:* FasterRisk model for the COMPAS dataset, predicting whether individuals are arrested within two years of release. Copy of Table 7. *Right:* Reduction to have relatively prime coefficients.

G.2 Transforming Features for Better Interpretability

Sometimes the original features are not as interpretable as they could be with some minor postprocessing. For example, Table 69 has features "previous case ≤ 10 or > 20" and "previous case ≤ 20". We can transform them into more interpretable and user-friendly features as "previous case ≤ 10", "$10 <$ previous case ≤ 20", and "previous case > 20". The transformed model is shown in Table 70.

1.	# of previous penal cases ≤ 2	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.633	1 point	+	...
4.	previous case ≤ 10 or > 20	-3 points	+	...
5.	previouse case ≤ 20	-5 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	14.9%	23.8%	35.8%	50.0%	64.2%	76.2%	85.1%

SCORE	-3	-2	-1	0	1	2
RISK	91.1%	94.8%	97.0%	98.3%	99.1%	99.5%

Table 69: Original FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years.

1.	# of previous penal cases ≤ 2	-2 points		...
2.	age in years ≤ 38.052	1 point	+	...
3.	age at first penal case ≤ 22.633	1 point	+	...
4.	previous case ≤ 10	-8 points	+	...
5.	$10 <$ previouse case ≤ 20	-5 points	+	...
6.	previouse case > 20	-3 points	+	...
		SCORE	=	

SCORE	-10	-9	-8	-7	-6	-5	-4
RISK	14.9%	23.8%	35.8%	50.0%	64.2%	76.2%	85.1%

SCORE	-3	-2	-1	0	1	2
RISK	91.1%	94.8%	97.0%	98.3%	99.1%	99.5%

Table 70: Postprocessed FasterRisk model for the Netherlands dataset, predicting whether defendants have any type of charge within four years. We have transformed the "previous case" feature for better interpretability. Note that in the original model, samples with previous case values less than 10 accumulate -8 points, -3 for the 4th line and -5 for the 5th line. In the transformed model, this case is more clearly stated in line 4.

H Discussion of Limitations

FasterRisk does not provide provably optimal solutions to an NP-hard problem, which is how it is able to perform in reasonable time. FasterRisk's models should not be interpreted as causal. FasterRisk creates very sparse, generalized, additive models, and thus has limited model capacity. FasterRisk's models inherit flaws from data on which it was trained. FasterRisk is not yet customized to a given application, which can be done in future work. We note that even if a model is interpretable, it can still have negative societal bias. (Generally, it is easier to check for such biases with scoring systems than with black box models). Looking at a variety of models from the diverse pool can help users to find models that are more fair.