Accelerated and interpretable oblique random survival forests:

Response to reviewers

We thank the reviewers and associate editor for their thoughtful and helpful comments. Below we provide a point-by-point response to each of their comments.

## Reviewer 1 Comments to the Author

This manuscript adds two contributions to the literature of random forests. 1) The development of a computational efficient oblique random forest algorithm for right censored/survival data and 2) An alternative approach to estimate variable important using what is called “negation variable importance”. The authors apply their approach on both simulated and real data sets. Overall, the manuscript is well written, and the manuscript is in line with JCGS’s scopes and aims. The authors do an excellent job outlining their approach and planned future directions.

Below are a few minor comments that I would like to have addressed.

1. The validity of the Cox partial likelihood function requires the non-informative censoring assumption (true survival time and censoring time are independent given the features X). I did not see this explicitly stated as an assumption in Section 2.3.

**Response**: We agree with the reviewer and have modified the manuscript to include this information. Specifically, because non-informative censoring presents a possible limitation for our approach, we have added the following text to Section 5.2:

“The validity of the Cox partial likelihood function depends on an assumption of non-informative censoring (*i.e.*, true survival time and censoring time are independent given the predictors). Thus, the oblique RSF may be improved by incorporating methods to account for informative censoring prior to identifying a linear combination of variables.”

1. For the simulations proposed in Section 3.2.3, the authors use the simsurv package. The underlying data generating model was not included so it is unclear how exactly the data were simulated. More clarity on the data generation procedure would be appreciated. I am guessing that the authors generated data under a PH model. However, it would be advantageous to see the performance of aosrf under data generated under a non-PH model.

**Response**: Thank you for noticing this. We have clarified the sentence in Section 3.2.3 so that the underlying model is clear. In the updated paper, we write:

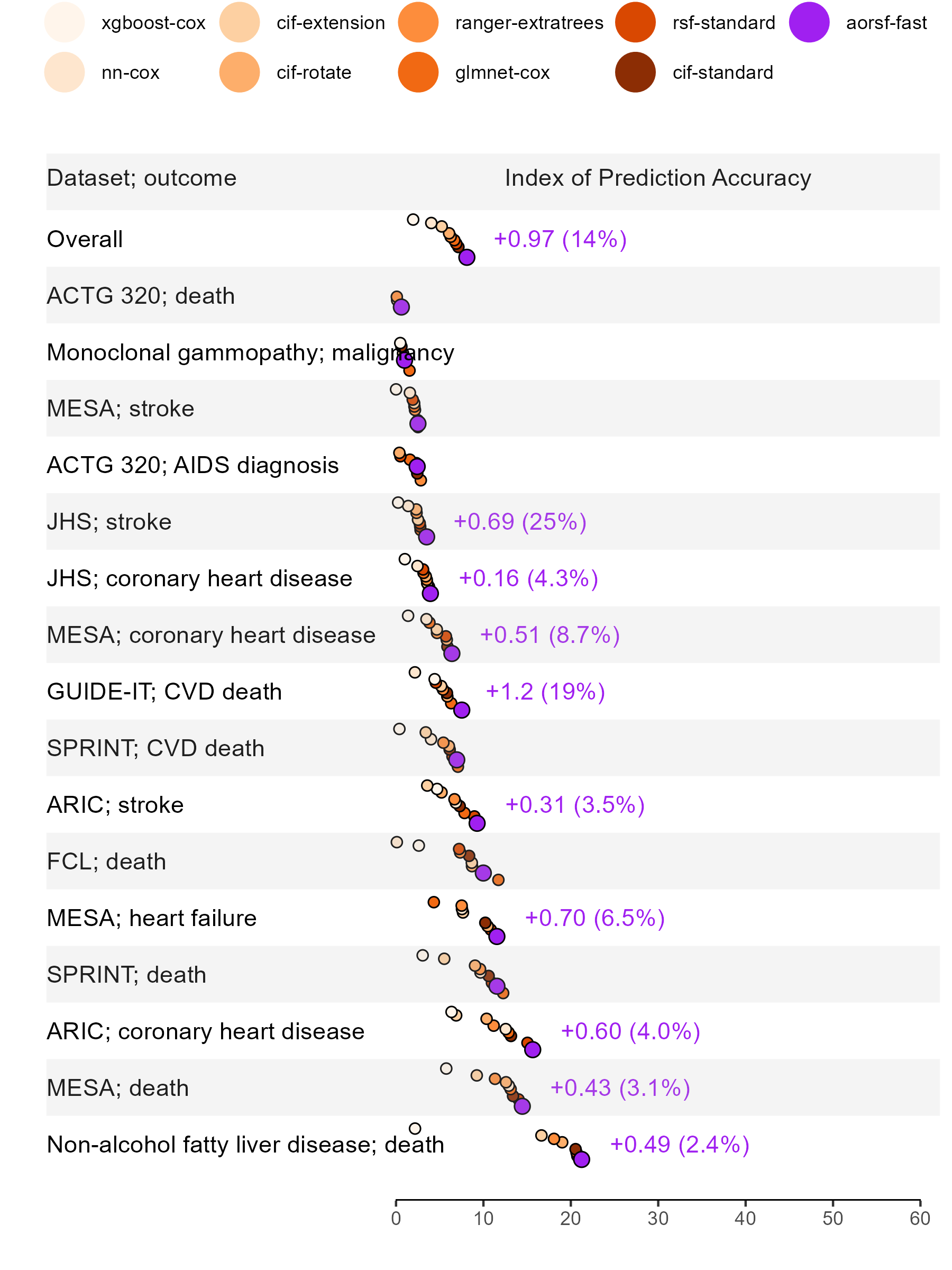
“A time-to-event outcome with roughly 45% of observations censored was generated using the simsurv package from a Weibull distribution”

We agree that a scenario where the PH assumption does not hold is interesting. In our prior paper introducing the original oblique RSF (PMC9875945), scenario C in our simulation was designed to investigate the prediction accuracy of oblique RSFs when the PH assumption was not valid. We found that the oblique RSF did well in this scenario (see Table 1 of PMC9875945), and thus we did not replicate the non-PH scenario in our current paper.

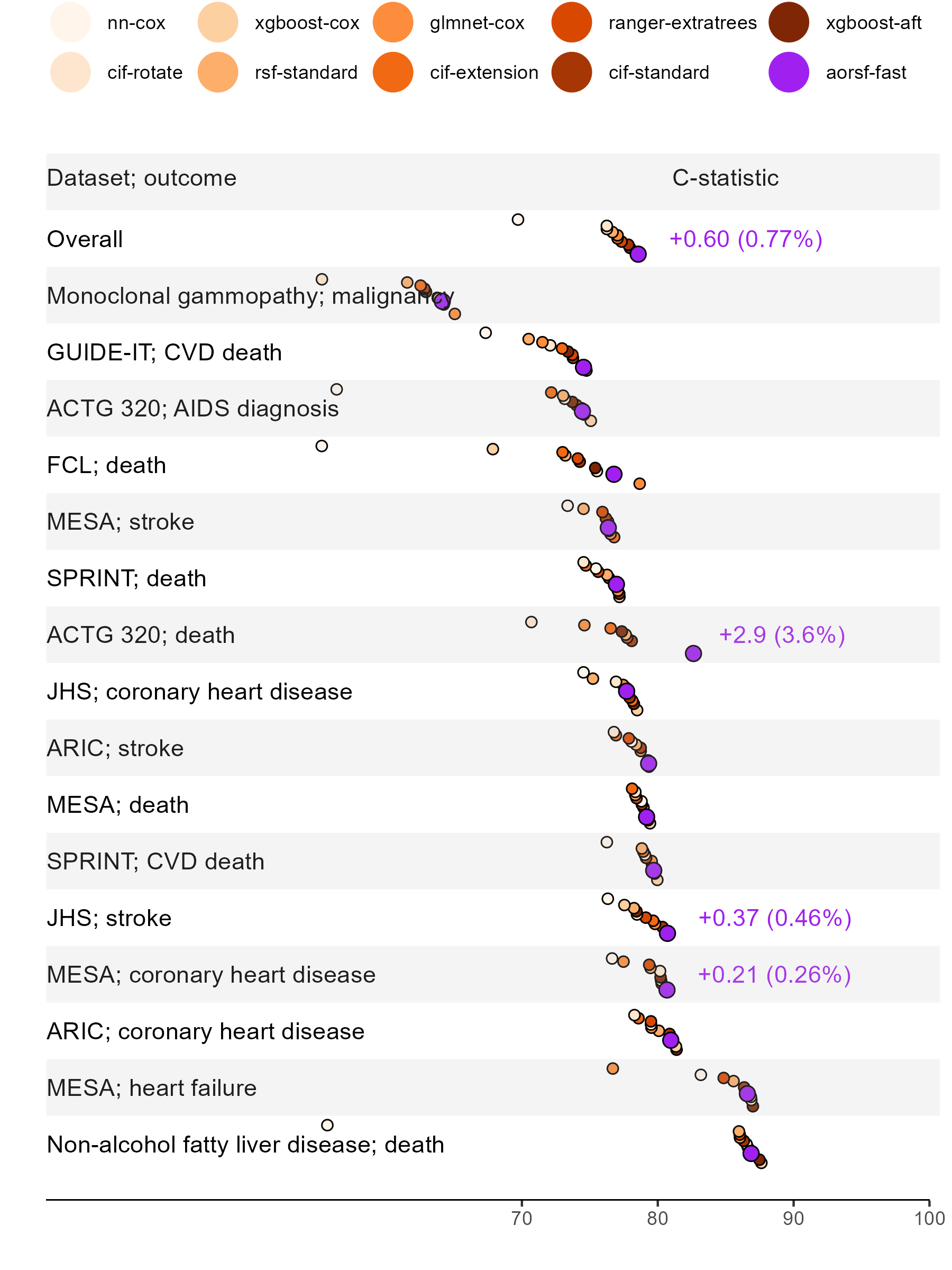
1. It would be good to see how the various methods perform under high censoring, especially since several of the datasets that are included have censoring > 80%.

**Response**: We agree that this is an important consideration. Figures 1 and 2 examine our learning algorithms only in the datasets with at least 80% censoring, and the results are similar to our main analysis. Due to space constraints, we have not included these results in our revision.

**Figure 1**: Index of prediction accuracy in risk prediction tasks with censoring of 80% or higher. Text appears in tasks where the accelerated oblique random survival forest obtained the highest score, showing absolute and relative improvement over the second best learner.



**Figure 2**: Time-dependent concordance statistic in risk prediction tasks with censoring of 80% or higher. Text appears in tasks where the accelerated oblique random survival forest obtained the highest score, showing absolute and relative improvement over the second best learner.



## Reviewer 2 Comments to the Author

1. Lines 42-49 on page 6: I would like to request further clarification on the process for obtaining linear combinations of predictor variables through the use of Newton-Raphson scoring.

**Response**: In each non-leaf node, the process for obtaining a linear combination of predictors is as follows:

*Step 1*. Identify a random subset of mtry predictors.

*Step 2*. Apply Newton-Raphson scoring to estimate as in Equation (1), using all predictors from Step 1.

*Step 3*. Create the linear combination of variables using the estimate from Step 2., i.e.,

*Step 4*. Identify a cut-point for that maximizes a splitting criterion.

We have added the following text to the first paragraph of page 7 in the revised manuscript to clarify Step 3:

“After obtaining , a linear combination of variables, , can be obtained by computing ”

This clarification is also included in line 14 of Algorithm 1 of the revised manuscript.

1. Equation in line 13 on page 7, I have some doubts as to whether the sign on the right-hand side should be negative (positive in Fisher scoring with the information matrix). Furthermore, it may be better to bold the beta symbol in the equation to be consistent with the previous formula.

**Response**: Thank you for noticing these details. We have updated this equation to match the terms given in Equation 3.8 in “Modeling Survival Data: Extending the Cox Model” By Terry M. Therneau and Patricia M. Grambsch. Specifically, in the revised manuscript, we write:

“As described in Therneau and Grambsch (2000), a vector of estimated regression coefficients, , is updated in each step of the procedure: where is the score vector and is the inverse of the observed information matrix.”

We have also written in bold throughout the revised manuscript.

## Associate Editor’s Comments to the Author:

This paper improves the computational efficiency and interpretation of oblique random survival forests (RSFs). The paper is clearly written. The numerical experiments are exceptionally well-planned, executed, and presented. Two reviewers’ and below comments (mostly minor) need to be addressed.

1. The package associated with this manuscript, aorsf, is one of the supported engines by tidymodels (<https://urldefense.com/v3/__https://parsnip.tidymodels.org/reference/details_rand_forest_aorsf.html__>;!!GA8Xfdg!yI2a4uGp0KH2vYEOZyxWCuA5MK9PUwHaWFlkvdEAr73zIbT1eicot7N-Mj6CaKHRP3OUkh030vi3ORph5zQfVgCSRYXfHA$ ) and is widely used. This paper has potentially high impact. The numerical experiments are extensive and well-planned. They set a high standard for JCGS articles. It boasts 21 data sets with a total of 35 learning tasks. Performances of learners are compared using the Bayesian linear mixed model.

**Response**: Thank you!

1. Conceptually, how does oblique RSF compare to first creating linear (PCA, PLS) and nonlinear features (UMP) and then applying standard axis RSF? Authors don’t need to make extra comparisons. A brief explanation suffices.

**Response**: This is an important distinction and we have incorporated it into the Introduction of the revised manuscript:

“Conceptually, oblique trees are similar to methods that use covariate rotation or extension to generate linear combinations of predictors prior to growing axis-based trees (Zhou et al., 2016; Wang and Zhou, 2017). The difference is that oblique trees generate linear combinations of predictors within each non-leaf node, using only the data and predictors associated with that node, instead of generating the linear combinations prior to growing the tree. This ``node-specific’’ approach to creating linear combinations of predictors leads to greater diversity in oblique tree ensembles, which may lead to greater prediction accuracy Breiman (2001).”

1. page 7, line 13: What are the and functions? Score and Hessian? Please define.

**Response**: We have clarified and (see response to comment #2 from Reviewer #2)

1. page 7, line 23 and footnote: should be .

**Response**: We have updated the text and footnote to correctly reflect that is transposed. We retained the hat over for consistency with our definition of and defined above.

1. Page 12, line 2: number and density of layers means number and width of layers or number and dropout rates of layers?

**Response**: We have updated this sentence to provide more clarity and detail on how we tuned neural networks: “For neural networks, the number of layers (*i.e.*, length) and number of nodes in the layers (*i.e.*, width) was tuned, while dropout rate was fixed at 10%, batch size was fixed at 32 observations, and the rectified linear unit activation function was applied. In addition, neural networks completed a maximum of 500 epochs, with possible early stopping based on prediction accuracy in a validation set comprising 25% of its training data.”

1. For boosting, is the learning rate tuned? How are the neural networks tuned (depth and width of layers, drop out rate, etc)?

**Response**: We have updated our description of tuning for neural networks (see previous response) and boosting. Our updated text for boosting is as follows: “Specifically, tuning for boosting models included identifying the number of steps to complete. The maximum number of steps was 5000, the learning rate was fixed at 0.01, and early stopping was applied if there was no improvement in cross-validated negative log-likelihood for 25 steps.”

1. In general, the paper lacks details on how several competing learning methods are tuned. For reproducibility, (1) how the data sets are retrieved and pre-processed and (2) the code and script for numerical experiments need to be made public.

**Response**: We have included more detail in the text regarding tuning of the competing learners (see response to comment 5 and 6), and have included details regarding tuning of competing learners in the ‘Description’ column of Table 1. We have included details on how each data set was retrieved and pre-processed in the ‘Data Sources’ section of the Appendix. We have made the code and script for numerical experiments publicly available at <https://github.com/bcjaeger/aorsf-bench>. In particular,

* data pre-processing code can be found here: <https://github.com/bcjaeger/aorsf-bench/blob/main/R/data_load.R>
* code for tuning of competing learners can be found here: <https://github.com/bcjaeger/aorsf-bench/blob/main/R/model_fit.R>

1. Please briefly explain what nested cross-validation is.

**Response**: We have included a brief explanation in the update: “Nested cross-validation includes an ‘inner’ cross-validation loop to evaluate different specifications of tuning parameters and an ‘outer’ loop that evaluates the tuned model, providing an unbiased estimate of the underlying model and its tuning procedure”