Supplementary Material for Estimating Calibrated Individualized Survival Curves with Deep Learning

Fahad Kamran,[1] **Jenna Wiens**[1]
[1] Computer Science and Engineering
University of Michigan
fhdkmrn, wiensj@umich.edu

Deep Survival Analysis Architectures

Recently, many have applied neural networks to data with censored individuals for survival analysis (Luck et al. 2017; Katzman et al. 2018; Ranganath et al. 2016; Alaa and van der Schaar 2017). However, many of these models rely on assumptions about the distributional form of the time-to-event data, such as the proportional hazards assumption (Cox 1972; Wang et al. 2017). These assumptions may not generalize to new data. Accordingly, we focus our analysis on deep survival analysis architectures that achieve state-of-the-art discriminative results without explicitly relying on any distributional assumptions. Despite reported gains in discriminative performance, to date, these models have not been evaluated in terms of calibration.

DeepHit was one of the first fully distribution-free methods for survival analysis (Lee et al. 2018). DeepHit corresponds to a feed-forward neural network architecture that takes as input an individual's covariates $\mathbf{x}_i$, and outputs a probability distribution $\hat{\mathbf{y}}_i \in [0,1]^\tau$, where $\hat{y}_{i,t}$ corresponds to the estimated $\hat{P}(Z = t|\mathbf{x}_i)$. The CIF at time t can then be estimated as $\hat{F}(t|\mathbf{x}_i) = \sum_{j=1}^{t} \hat{y}_{i,j}$. The final layer of DeepHit is a softmax output layer requiring $\hat{F}(\tau|\mathbf{x}_i) = 1$. This formulation assumes that, by the end of the time horizon τ, every individual will have had the event. Hence, this formulation will incorrectly estimate the true underlying survival process for individuals who survive beyond time τ. Moreover, as DeepHit outputs a fixed-sized vector, it can not be used to forecast survival curves past the specified time-horizon τ.

DRSA, or deep recurrent survival analysis, alleviates this structural issue of DeepHit while taking advantage of the sequential patterns present in survival analysis (Ren et al. 2019). DRSA uses a long short-term memory (LSTM) network that takes as input at timestep t, a concatenation of an individual's covariates $\mathbf{x}_i$ and t (Hochreiter and Schmidhuber 1997). The output of the LSTM at time t is passed into a fully connected layer with a sigmoid activation function that outputs $\hat{\lambda}(t|\mathbf{x}_i)$. Accordingly, we can estimate the survival probability at timestep t as $\hat{S}(t|\mathbf{x}_i) = \prod_{j:j\leq t}(1 - \hat{\lambda}(j|\mathbf{x}_i))$, and the probability of the event occurring at timestep t as $\hat{P}(Z = t|\mathbf{x}_i) = \hat{\lambda}(t|\mathbf{x}_i)\prod_{j<t}(1 - \hat{\lambda}(j|\mathbf{x}_i))$. Since DRSA does not make assumptions about the probability of survival at the end of the horizon while still allowing for variable-length forecasting of survival curves, we build on this architecture in our proposed approach.

Full Proof that $\mathcal{L}_{RPS}$ Elicits Calibrated Survival Curves

Claim. *Training deep survival models using $\mathcal{L}_{RPS}$ will result in well-calibrated estimates of survival.*

Proof. Consider n individuals with identical or near-identical covariates with observed event times $\{z_i\}_{i=1}^n$. Define the counting-based Kaplain-Meier estimate for these individuals at time t as $KM_t^n = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{t<z_i}$, where $\lim_{n\to\infty} KM_t^n$ is the underlying survival probability at time t for these n individuals.

A survival model will estimate one survival probability for these n individuals at time t. Define this value as $\hat{p}_t$. A well-calibrated survival model will output a $\hat{p}_t$ that closely aligns with the underlying survival probability $\lim_{n\to\infty} KM_t^n$. Consider the optimization problem of finding $\hat{p}_t$ which will minimize $\mathcal{L}_{RPS}$. This problem can formally be set-up as $\arg\min_{\hat{p}_t}\sum_{i=1}^n(\hat{p}_t - \mathbb{1}_{t<z_i})^2$.

First, this optimization problem is strictly convex and has a unique minimum, as the second derivative is positive everywhere. such that any minimizer must be the unique minimizer to this loss function. In order to do so, consider taking the second derivative of the objective function with respect to $\hat{p}_t$.

$$\frac{\partial^2}{\partial \hat{p}_t^2}\left(\sum_{i=1}^n(\hat{p}_t - \mathbb{1}_{t<z_i})^2\right) =$$

$$\frac{\partial}{\partial \hat{p}_t}\left(2\hat{p}_t - \frac{2}{n}\sum_{i=1}^n \mathbb{1}_{t<z_i}\right) =$$

$$2 \geq 0$$

To find the value of $\hat{p}_t$ that minimizes this objective function ($\hat{p}_t^*$), we set the derivative equal to zero.

$$\frac{\partial}{\partial \hat{p}_t^*}\left(\sum_{i=1}^{n}(\hat{p}_t^* - \mathbb{1}_{t<z_i})^2\right) = 0$$

$$2\hat{p}_t^* - \frac{2}{n}\sum_{i=1}^{n}\mathbb{1}_{t<z_i} = 0$$

$$\hat{p}_t^* = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{t<z_i}$$

The unique estimated survival probability that minimizes the objective function is equivalent to the average survival status for all n individuals at time t. This unique minimum is equal to KM_t^n which, as n gets large, is equal to the true underlying survival probability for these individuals at time t. Hence, training a survival model to minimize $\mathcal{L}_{RPS}$ will result in estimated survival probabilities that align well with the true survival probabilities. □

Censored DDC

In the case of censored individuals, we only know that prior to censoring the event did not occur. Following the probability integral transform argument used to justify DDC, for a well-calibrated model, we would expect half of the individuals to have the event after reaching an estimated survival probability of 50%. If *more* than half the individuals are censored after reaching an estimated survival probability of 50%, then we can conclude that the model is *not* well-calibrated. However, if *less* than half of the individuals are censored after reaching an estimated survival probability of 50%, we cannot conclude anything with respect to model calibration (the event may take place at any time after censoring). Given these limitations, without strong assumptions on the event time distribution for censored individuals, one cannot make meaningful conclusions regarding the calibration of a model for censored individuals. To this end, while we measure discriminative performance across both uncensored and censored individuals, we focus our evaluation of calibration on uncensored individuals.

Trade-Off Between Discriminative Performance and Calibration

To display the trade-off between discriminative performance and calibration, we simulate 1,000 covariates and corresponding sampled event times through the following scheme:

$$\mathbf{X} = (\mathbf{X}^a, \mathbf{X}^b)^T \in \mathbb{R}^{1,000\times 20}$$
$$\mathbf{X}^a = (\mathbf{X}_1^a, \mathbf{X}_2^a) \in \mathbb{R}^{500\times 20}$$
$$\mathbf{X}^b = (\mathbf{X}_1^b, \mathbf{X}_2^b) \in \mathbb{R}^{500\times 20}$$
$$\mathbf{X}_1^a, \mathbf{X}_1^b \sim U(0,10)^{10}$$
$$\mathbf{X}_2^a \sim U(10,20)^{10}$$
$$\mathbf{X}_2^b \sim U(5,15)^{10}$$
$$z_i \sim LN(.5(\mathbf{1}^T\mathbf{x}_i^{1:10})^2 + 2(\mathbf{1}^T\mathbf{x}_i^{11:20})^2, 0.5)$$

Note that U and LN denote a uniform and a log-normal distribution respectively. We consider τ (the time-horizon) to be the 50th percentile of sampled event times, in order to right-censor half of the individuals. Finally, we place all time to events into one of 100 equally spaced time bins.

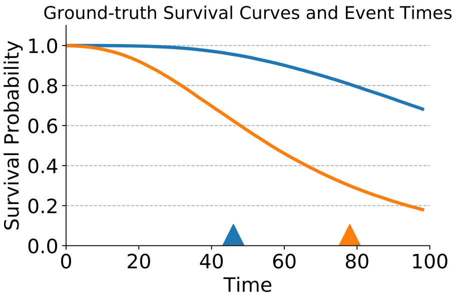

Figure 1: An example pair of ground-truth survival curves for 2 individuals from a simulated stochastic process; see the Appendix for more details. Triangles denote the observed event times. As the blue individual experienced the event at a high survival probability, they will consistently be ranked incorrectly when compared to other individuals who have a lower survival probability but experience the event later (*e.g.*, the orange individual). These examples will contribute negatively to the C-index evaluation, despite good calibration.

Given this simulation, we calculate the C-index value for the ground-truth log-normal survival curves. The average C-index of the ground-truth survival curves in these finite samples across 1000 replications of the simulation is .760 (95% Confidence Interval: (.742, .778)). This is due to examples such as the one displayed in the **Figure 1**. Though an individual can experience an event early, it is not necessarily true that their true survival probability is low. These situations result in incorrect rankings among different individuals, which contributes negatively towards the C-index value.

Importantly, we note that this is due to the single sample definition of discrimination. For example, for a particular observed outcome distribution, it is possible to achieve perfect discrimination (as measured by the C-index) by estimating heaviside distributions that drop to 0 at the observed event times. However, these distributions do not take into account the stochasticity that likely exists in the survival process. Due to this stochasticity, it is unlikely for the underlying survival curves to provide perfect discriminative performance (i.e. a C-index of 1) with respect to the observed outcomes, showing an important trade-off that is necessary to consider when evaluating survival models.

Additional Experimental Set-Up Details

Dataset Details We consider two public clinical datasets: the Northern Alberta Cancer Dataset and the CLINIC dataset. For each dataset, we use the same 60/20/20% train/validation/test split across model initializations in order to train and evaluate our models. We stratify our random splits in order to ensure a roughly equal proportion of censored individuals in each split. We normalize all covariates by the mean and standard deviation of each feature in the

Table 1: Discriminative (C-index) and calibration performance (DDC, D-Calibration, Averaged Brier Score), as well as the trade-off between the two (total score) for the NACD and CLINIC datasets (mean ± standard deviation across random initializations, number of times passing the statistical test for D-Calibration). Lower DDC and Brier score values indicate better performance, while higher values of C-index, D-Calibration, and total score indicate better performance. The proposed training approach consistently leads to improvements in calibration, without sacrificing discriminative performance or Brier score. An * indicates results that are statistically significant over all baselines using a paired t-test ($p < .05$).

Model	**NACD**					**CLINIC**				
	C-index ↑	**DDC** ↓	**D-Calibration** ↑	$\overline{\textbf{Brier}}$ ↓	**Total Score** ↑	**C-index** ↑	**DDC** ↓	**D-Calibration** ↑	$\overline{\textbf{Brier}}$ ↓	**Total Score** ↑
Ren et al. 2019	$.748 \pm .002$	$.025 \pm .012$	1	$.101 \pm .002$	$.846 \pm .004$	$.616 \pm .003$	$.138 \pm .002$	0	$.107 \pm .000$	$.719 \pm .003$
MTLR	$.750 \pm .000$	$.062 \pm .000$	0	$.101 \pm .000$	$.834 \pm .000$	$.608 \pm .000$	$.168 \pm .000$	0	$.106 \pm .000$	$.702 \pm .000$
DeepHit ($\mathcal{L}_{log}$)	$.751 \pm .002$	$.083 \pm .005$	0	$.102 \pm .000$	$.826 \pm .003$	$.616 \pm .003$	$.133 \pm .004$	0	$.103 \pm .000$	$.720 \pm .002$
DeepHit ($\mathcal{L}_{log} + \lambda\mathcal{L}_{kernel}$)	$.748 \pm .004$	$.020 \pm .005$	0	$.107 \pm .001$	$.849 \pm .003$	$.624 \pm .001$	$.063 \pm .007$	0	$.106 \pm .001$	$.749 \pm .002$
Proposed - $\mathcal{L}_{RPS}$	$.741 \pm .008$	$.305 \pm .089$	0	$.207 \pm .034$	$.715 \pm .050$	$.628 \pm .003$	$.241 \pm .022$	0	$.153 \pm .002$	$.687 \pm .011$
Proposed - $\mathcal{L}_{kernel}$	$.742 \pm .003$	$.012 \pm .002$	3	$.101 \pm .003$	$.847 \pm .001$	$.615 \pm .005$	$.097 \pm .006$	0	$.110 \pm .001$	$.731 \pm .005$
Proposed Method	$.742 \pm .006$	$\mathbf{.007 \pm .003}^*$	**5**	$.104 \pm .002$	$.850 \pm .003$	$.627 \pm .001$	$.056 \pm .011$	0	$.106 \pm .001$	$.753 \pm .004$

training set.

Additional Baselines. For completeness, we report the results for two additional baseline methods. Namely, we train two variants of the feed-forward DeepHit model. First, we train the DeepHit architecture with the loss as it was originally proposed ($\mathcal{L}_{log} + \lambda\mathcal{L}_{kernel}$). To examine the importance of $\mathcal{L}_{kernel}$ in DeepHit and examine the performance of $\mathcal{L}_{log}$ alone, we also consider evaluating the performance of DeepHit without the kernel loss ($\lambda = 0$).

Additional Training and Hyperparamter Details. All DRSA models had the same architecture: a one-layer LSTM with hidden size 100 and a single feed-forward layer with a sigmoid activation on the output for each time-step. For DeepHit, we followed the same architecture proposed in the original paper. We considered learning rates of *1e-3* and *1e-4*, but preliminary results found no comparable difference in performance on the held-out validation set, so we continued using a learning rate of *1e-3*. In order to tune the σ hyperparameter for the $\mathcal{L}_{kernel}$ loss function, we considered σ values from 0.1 to 10. σ was then chosen based on performance on the held-out validation set on the NACD dataset. This optimal σ value ($\sigma = .8$) was used for both the NACD dataset and the CLINIC dataset in order to test generalizability of the relationship between $\mathcal{L}_{RPS}$ and $\mathcal{L}_{kernel}$ in the composite loss. Other hyperparameters, such as the weighting scheme used in conjunction with $\mathcal{L}_{RPS}$ due to the right-skewed time-to-event distribution, were chosen based on performance on the held-out validation set as well.

In order to tune regularization constants of MTLR, which control the amount of smoothing for the model, we used the cross-validation scheme built into the MTLR R package.

Additional Results

The proposed method continues to consistently outperforms all baselines with respect to DDC and D-calibration, while maintaining comparable C-index and average Brier score values (**Table 1**). Compared to DRSA and DeepHit with $\lambda = 0$, the proposed method results in a statistically significant improvement in calibration across both tasks (NACD DDC: .025 and .083 vs. .007, CLINIC DDC: .138 and .133 vs .056). This improvement, however, is accompanied by a small decrease in C-index in the NACD dataset. Moreover, training using $\mathcal{L}_{RPS}$ alone results in better calibration than both DRSA and DeepHit trained using only $\mathcal{L}_{log}$ (NACD DDC: .025 and .083 vs .012, CLINIC DDC: .138 and .133 vs .097), with minimal drops in discriminative performance. These empirical results support the original hypothesis that training using $\mathcal{L}_{RPS}$ should result in survival models that better balance discriminative performance and calibration.

DeepHit that includes training with $\mathcal{L}_{kernel}$ consistently results in better calibration compared to DeepHit without this loss function (DeepHit ($\lambda = 0$)). This supports the hypothesis that $\mathcal{L}_{kernel}$ can act as a scaling mechanism to calibrate survival estimates without sacrificing discriminative performance. Despite this increased performance, our proposed approach still achieves better calibration performance (NACD DDC: .020 vs .007, CLINIC DDC: .063 vs .057), while also maintaining a better trade-off between calibration and discriminative performance, as shown through the total score.

Overall, these results continue to support our original hypothesis regarding the efficacy of the training scheme. We show that training using $\mathcal{L}_{RPS}$ outperforms models that solely train using $\mathcal{L}_{log}$, while including the kernel loss function can consistently improve calibration performance with respect to DDC and D-Calibration. Finally, the best performance consistently comes from our proposed method, the combination of $\mathcal{L}_{RPS}$ and $\mathcal{L}_{kernel}$.

References

Alaa, A. M.; and van der Schaar, M. 2017. Deep multi-task gaussian processes for survival analysis with competing risks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2326–2334. Curran Associates Inc.

Cox, D. R. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34(2): 187–202.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Katzman, J. L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; and Kluger, Y. 2018. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology* 18(1): 24.

Lee, C.; Zame, W. R.; Yoon, J.; and van der Schaar, M. 2018. Deephit: A deep learning approach to survival analysis with competing risks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Luck, M.; Sylvain, T.; Cardinal, H.; Lodi, A.; and Bengio, Y. 2017. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245* .

Ranganath, R.; Perotte, A.; Elhadad, N.; and Blei, D. 2016. Deep survival analysis. *arXiv preprint arXiv:1608.02158* .

Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; Qiu, L.; and Yu, Y. 2019. Deep Recurrent Survival Analysis. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

Wang, L.; Li, Y.; Zhou, J.; Zhu, D.; and Ye, J. 2017. Multitask survival analysis. In *2017 IEEE International Conference on Data Mining (ICDM)*, 485–494. IEEE.