
Survival Regression with Proper Scoring Rules and Monotonic Neural Networks

David Rindt*

Robert Hu*

David Steinsaltz

Dino Sejdinovic

University of Oxford

Abstract

We consider frequently used scoring rules for right-censored survival regression models such as time-dependent concordance, survival-CRPS, integrated Brier score and integrated binomial log-likelihood, and prove that neither of them is a proper scoring rule. This means that the true survival distribution may be scored worse than incorrect distributions, leading to inaccurate estimation. We prove that, in contrast to these scores, the right-censored log-likelihood is a proper scoring rule, i.e., the highest expected score is achieved by the true distribution. Despite this, modern feed-forward neural-network-based survival regression models are unable to train and validate directly on the right-censored log-likelihood, due to its intractability, and resort to the aforementioned alternatives, i.e., non-proper scoring rules. We therefore propose a simple novel survival regression method capable of directly optimizing log-likelihood using a monotonic restriction on the time-dependent weights, coined SurvivalMonotonic-net (SuMo-net). SuMo-net achieves state-of-the-art log-likelihood scores across several datasets with 20–100× computational speedup on inference over existing state-of-the-art neural methods, and is readily applicable to datasets with several million observations.

1 INTRODUCTION AND RELATED WORKS

Survival analysis is a class of statistical methods for analyzing the time until the occurrence of an event, such as death in biological organisms (Ziehm and Thornton, 2013) or failure in mechanical systems (Susto et al., 2015). We focus on survival regression, where we relate the time that passes before some event occurs to one or more covariates that may be associated with that quantity of time. In particular, we consider the case in which the event times are right-censored, meaning that for some individuals it is known their event happened after a certain time, but not exactly when. For example, a patient who is still alive when the study ends would be right censored. Applications of survival regression on right-censored data are numerous, with the most classical example being the study of recovery time from medical treatments (Laurie et al., 1989). Other applications include modeling the time until the default of a business (Dirick et al., 2017), and churn prediction (Van den Poel and Lariviere, 2004).

Related Work Classical models for survival regression include the Cox proportional hazards model (Cox, 1972), the accelerated failure time (MacKenzie, 1982), as well as various parametric models (Kleinbaum and Klein, 2005). Over the last decade, many machine learning methods for survival data have been proposed. We here mention the most relevant methods which allow more flexible survival functions than the Cox model, and refer to Kvamme et al. (2019) for further references. In Katzman et al. (2018) the Cox model is extended by replacing the linear function with a neural network. Kvamme et al. (2019) further extends on this idea and includes the survival time to the neural network, achieving state-of-the-art performance. In Ishwaran et al. (2008) a random-forest approach is taken, in which cumulative hazard functions are computed using the Nelson–Aalen estimator. In Lee et al. (2018) a discrete distribution is learned by optimizing a mix of the discrete likelihood and a rank-based score. Ranganath et al. (2016) and Miscouridou et al. (2018) first learn a latent representation of the covariates, and then

*Equal contribution, author order decided by coin flip.
Correspondence to: robert.hu@stats.ox.ac.uk.

learn a survival function using the latent representation. In particular, the latter method can deal with missing data in the context of electronic health records. In Chapfuwa et al. (2018) an adversarial model is used.

Intractable Likelihood As we model a survival *distribution*, it is natural to consider approaches based on maximizing the likelihood. In the particular case of right-censoring, we later prove that the right-censored likelihood actually is a proper scoring rule (Gneiting and Raftery, 2007), meaning that the highest expected score is obtained by the true distribution. However, none of the survival regression methods overviewed above have a tractable likelihood, impeding its direct use in estimation. In particular, Kvamme et al. (2019); Katzman et al. (2018) and Ishwaran et al. (2008) all learn discrete distributions, consisting of point masses at fixed times. In Lee et al. (2018), time is discretized into bins and while Ranganath et al. (2016), and Miscouridou et al. (2018) take a Bayesian approach, the resulting posterior distribution is also intractable. In each case computing the likelihood of new observations is problematic, as no method has direct access to the cumulative survival distribution. Some recent work by Groha et al. (2021) and Tang et al. (2020) proposes NeuralODEs to model the right-censored survival likelihood directly as a differential equation in time. While NeuralODEs are a very flexible class of models, they instead have pathologies relating to scalability (Dupont et al., 2019), since they scale polynomially in the number of parameters used (Massaroli et al., 2021).

Alternative scoring rules While the likelihood is a standard measure of model fit for uncensored data, due to its intractability in right-censored data one typically resorts to alternative scoring rules. The most commonly used scores to evaluate the fit of a right-censored regression model are the time-dependent concordance (Antolini et al., 2005), the survival-CRPS (Avati et al., 2020), the Brier score for right-censored data (Graf et al., 1999) and the binomial log-likelihood. In this paper we explore pathologies of these alternative scoring rules and propose *SurvivalMonotonic neural net* (SuMo-net) as a scalable and flexible survival regression model that optimizes likelihood directly. We summarize our contributions as:

1. We show theoretically and experimentally that time-dependent concordance, (integrated) Brier score and binomial log-likelihood, survival-CRPS are not proper scoring rules, meaning inaccurate distributions may achieve better scores than the true distribution.
2. We give a simple proof that right-censored log-likelihood is a proper scoring rule.
3. We introduce a novel survival regression model which directly optimizes right-censored log-likelihood and compare it to the existing methods in terms of performance and scalability, achieving state-of-the-art results on a range of datasets, together with 20–100 times speedups. Codebase: <https://github.com/MrHuff/Sumo-Net.git>

The paper is organized as follows: In Section 2 we introduce the notation used for survival regression. In Section 3, we define what a proper score is and show that time-dependent concordance, (integrated) Brier score and binomial log-likelihood, survival-CRPS are not proper scores. We further prove that the right-censored likelihood is a proper score. *SurvivalMonotonic* neural net is introduced in Section 4. Section 5 provides extensive experiments. We conclude our work in Section 6.

2 BACKGROUND AND NOTATION

We are interested in an event-time $T \in \mathbb{R}_{\geq 0}$ and how it depends on a covariate $X \in \mathbb{R}^p$ for $p \geq 1$. We consider the case in which T is subject to right-censoring, where the event-time T is not known for every individual in the sample. Instead of observing T directly, we observe $Z = \min\{T, C\}$ where $C \in \mathbb{R}_{\geq 0}$ is a censoring time, as well as the indicator $D = 1\{Z = T\}$ indicating if we observe the event-time. A sample of size n can thus be denoted by $\{(X_i, Z_i, D_i)\}_{i=1}^n$, where we use the uppercase letters when we treat the dataset as random and lowercase letters otherwise. We let $S(t|x) := 1 - F(t|x) = \mathbb{P}(T > t|X = x)$ denote the (conditional) survival distribution, and, assuming T has a density, we let $f(t|x) := \frac{d}{dt}F(t|x)$ denote the density of the event-time.

The goal of this work is the following: If we let $S(t|x)$ denote the survival function of T , our aim is to estimate S with $\hat{S}(t|x)$ based on the sample $\{(X_i, Z_i, D_i)\}_{i=1}^n$. We also look at ways to assess if \hat{S} is an accurate estimate of the true distribution S .

In survival analysis one typically assumes that censoring is not informative about the event-time conditional on the covariate, which we formally state in the following assumption.

Assumption 1. (*Independent censoring*) We assume that conditionally on the covariate, the censoring- and event-time are independent. That is, we assume $T \perp\!\!\!\perp C|X$.

Let $S(t|x)$ be a survival function and let $f(t|x) = -\partial S(t|x)/\partial t$. Then under Assumption 1, the right-censored log-likelihood is defined as:

$$\log L_n = \sum_{i=1}^n d_i \log f(z_i|x_i) + (1 - d_i) \log S(z_i|x_i). \quad (1)$$

Right-censored survival regression

$$\begin{cases} F(t|x) = \int_0^t f(t'|x)dt' \\ S(t|x) = 1 - F(t|x) \\ S_Z(t|x) = 1 - \int_0^t f(t'|x)1_{t' < Z}dt' \end{cases}$$

Observed event-time	t_i	$\mathbb{R}_{\geq 0}$
Censoring time	c_i	$\mathbb{R}_{\geq 0}$
Right-censored times	$z_i = \min\{t_i, c_i\}$	$\mathbb{R}_{> 0}$
Indicator of observed event-time	d_i	$\{0, 1\}$
Covariates	x_i	\mathbb{R}^d

Both the density- and the survival function imply that, if one models the density, then the likelihood contains n integrals which need to be estimated, making it a difficult optimization objective (see e.g. p. 288 of Gu (2013)). For that reason many methods optimize the partial likelihood instead of the right-censored likelihood. (Cox, 1972).

3 EVALUATION CRITERIA AND PROPER SCORING RULES

A scoring rule \mathcal{S} takes a distribution S over some set \mathcal{Y} and an observed value $y \in \mathcal{Y}$ and returns a score $\mathcal{S}(S, y)$, assessing how well the model predicts the observed value. For a positive scoring rule, a higher score indicates a better model fit. A scoring rule is called proper if the true distribution achieves the optimal expected score, i.e., if

$$\mathbb{E}_{y \sim S} \mathcal{S}(S, y) \geq \mathbb{E}_{y \sim \hat{S}} \mathcal{S}(\hat{S}, y) \quad \text{for all distributions } \hat{S}.$$

In the context of survival regression, we call the scoring rule \mathcal{S} proper if for every true distribution S , every censoring distribution C and for every covariate x it holds that

$$\mathbb{E}_{T, C | X=x} \mathcal{S}(S(t|x), (Z, D)) \geq \mathbb{E}_{T, C | X=x} \mathcal{S}(\hat{S}(t|x), (Z, D))$$

for every distribution $\hat{S}(t|x)$, where in the above expression $T|X = x \sim S(t|x)$. By taking the expectation with respect to X in the above expression, we find that if \mathcal{S} is a proper score, then also

$$\mathbb{E}_{T, C, X} \mathcal{S}(S(t|X), (Z, D)) \geq \mathbb{E}_{T, C, X} \mathcal{S}(\hat{S}(t|X), (Z, D))$$

for every family of distributions $\hat{S}(t|x)$ with $x \in \mathbb{R}^p$.

3.1 Existing evaluation metrics are not proper scoring rules

We now show that time-dependent concordance (Antolini et al., 2005), the Brier score for right-censored data (Graf et al., 1999), the binomial log-likelihood and the survival-CRPS (Avati et al., 2020) are not proper scoring rules. For time-dependent concordance and the survival-CRPS, these are a novel results. Time-dependent concordance is not to be confused with

(Blanche et al., 2018), which considers *time-independent* concordance.

Time-dependent concordance Let (X, Z, D) and (X', Z', D') be i.i.d. covariates and right-censored event-times. Time-dependent concordance $\mathcal{S}_{C^{\text{td}}}$ (Antolini et al., 2005) is defined by

$$\begin{aligned} & \mathbb{P}(\hat{S}(Z|X) < \hat{S}(Z|X') | Z \leq Z', D = 1) \\ & + \frac{1}{2} \mathbb{P}(\hat{S}(Z|X) = \hat{S}(Z|X') | Z \leq Z', D = 1) \end{aligned}$$

where ties in the survival probabilities are dealt with as proposed in Ishwaran et al. (2008) (Step 3 of Section 5.1). It has been remarked that poorly calibrated models can still have a high time-dependent concordance (Kvamme et al., 2019). Rizopoulos et al. (2014) and Antolini et al. (2005) have pointed out that censoring affects time-dependent concordance. We now show that time-dependent concordance is not a proper score by constructing an example where a false distribution has higher score than the true distribution.

Example: optimizing concordance for a binary covariate. Let $X, X' \sim \text{Bernoulli}(1/2)$ independently. Then $\hat{S}(\cdot|\cdot)$ maximizes $\mathcal{S}_{C^{\text{td}}}$ if and only if it maximizes

$$\begin{aligned} & \mathbb{P}(\hat{S}(Z|X) < \hat{S}(Z|X'), Z \leq Z', D = 1 | X = 1, X' = 0) + \\ & \mathbb{P}(\hat{S}(Z|X) < \hat{S}(Z|X'), Z \leq Z', D = 1 | X = 0, X' = 1) = \\ & \int_0^\infty (1\{\hat{S}(t|1) < \hat{S}(t|0)\} f_T(t|1) S_C(t|1) S_T(t|0) S_C(t|0) \\ & + 1\{\hat{S}(t|0) < \hat{S}(t|1)\} f_T(t|0) S_C(t|0) S_T(t|1) S_C(t|1)) dt. \end{aligned}$$

This expression is optimized by constructing $\hat{S}(\cdot|\cdot)$ such that $\hat{S}(t|1) < \hat{S}(t|0)$ if and only if

$$f_T(t|1) S_C(t|1) S_T(t|0) S_C(t|0) \geq f_T(t|0) S_C(t|0) S_T(t|1) S_C(t|1) \iff \lambda_T(t|1) \geq \lambda_T(t|0)$$

where $\lambda_T(t|x) = f_T(t|x)/S_T(t|x)$ is called the hazard rate. Hence, in this example, to maximize concordance we need the ordering of $\hat{S}(t|1)$ and $\hat{S}(t|0)$ to be the reversed order of the true hazard rates. In Figure 1, we use this insight to generate a false distribution with concordance much higher than the concordance of the true distribution (Example 1). See Appendix 1.2 for details.

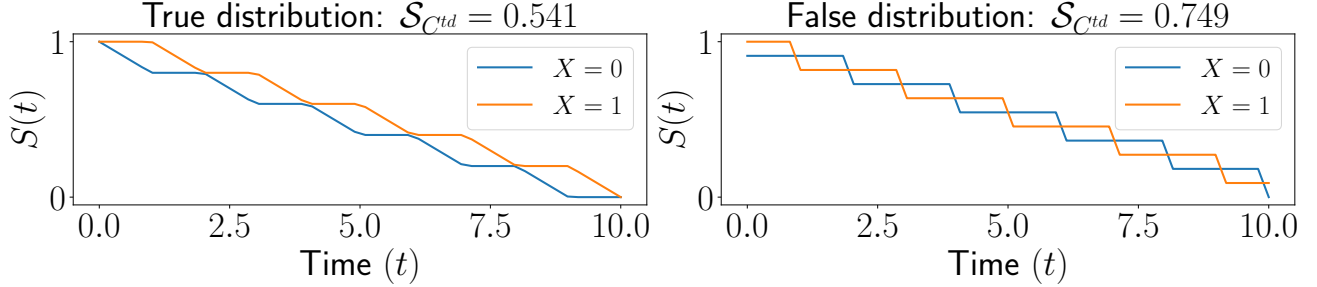


Figure 1: The survival curves for the true distribution on the left and a false distribution, designed to optimize concordance, on the right. By drawing a sample of size 1000 from the true distribution, we find that the concordance score of the true distribution is lower than the score of the false distribution, confirming that an inaccurate distribution can have high concordance.

	Example 1		Example 2					Example 3	
	$\mathcal{S}_L \uparrow$	$\mathcal{S}_{C^{td}} \uparrow$	$\mathcal{S}_L \uparrow$	$\mathcal{S}_{\text{Brier}}^{t=4.0} \downarrow$	$\mathcal{S}_{\text{BLL}}^{t=4.0} \uparrow$	$\mathcal{S}_{\text{IBS}} \downarrow$	$\mathcal{S}_{\text{IBLL}} \uparrow$	$\mathcal{S}_L \uparrow$	$\mathcal{S}_{\text{CRPS}} \downarrow$
True distribution score	-1.375	0.541	-1.795	0.220	0.085	0.032	-0.009	-0.513	3.884
Fake distribution score	-11.029	0.749	-1.820	0.204	0.122	0.031	-0.006	-0.626	1.646
Consistent?	✓	✗	✓	✗	✗	✗	✗	✓	✗

Table 1: Scores for true and fake distribution respectively. Bottom line indicates if the true distribution scores better than the fake distribution. Right-censored likelihood consistently scores the true distribution higher than a fake distribution, in contrast to non-proper scores.

Brier score The Brier score at time t for the estimated survival distribution $\hat{S}(\cdot|\cdot)$ is typically defined as (Graf et al., 1999)

$$\mathcal{S}_{\text{Brier}}^t(\hat{S}(t|x), (z, d)) = \frac{\hat{S}(t|x)^2 \mathbb{1}\{z \leq t, d = 1\}}{\hat{G}(z)} + \frac{(1 - \hat{S}(t|x))^2 \mathbb{1}\{z > t\}}{\hat{G}(t)}$$

where \hat{G} is the Kaplan-Meier estimate of the censoring distribution. Assuming that the estimated survival function of the censoring time C , $\hat{G}(\cdot)$ is the exact marginal distribution $G(\cdot) = \mathbb{P}(C > \cdot)$, then the expectation, conditional on the X_i equals

$$\begin{aligned} \mathbb{E}[\mathcal{S}_{\text{Brier}}(\hat{S}(t|x), (Z, D)) | X = x] &= \\ \hat{S}(t|x)^2 \int_0^t \frac{G(z|x)}{G(z)} f_T(z|x) dz & \quad (2) \\ + (1 - \hat{S}(t|x))^2 \frac{G(t|x)}{G(t)} S_T(t|x). & \quad (3) \end{aligned}$$

If $C \perp\!\!\!\perp X$, and $G(\cdot|x) = G(\cdot)$, for all x , then it is easy to see the above equals

$$\hat{S}(t|x)^2 (1 - S_T(t|x)) + (1 - \hat{S}(t|x))^2 S_T(t|x).$$

By setting $a_i = \hat{S}(t|x)$ and optimizing with respect to each of the a_i , we find that the expected Brier score is minimized for $\hat{S}(t|x) = S_T(t|x)$, i.e., when the estimated survival probabilities equal the true survival probabilities. *Under the assumption of censoring independent of the covariate and a perfectly estimated censoring distribution*, the Brier score is thus a proper score. However, as Equation 2 makes clear, the Brier

score may not be a proper score when $C \not\perp\!\!\!\perp X$. The problematic assumption on the censoring distribution has, for example, been reported in Kvamme and Borgan (2019). We construct an example (Example 2) where an inaccurate distribution achieves a lower (i.e., better) Brier score than the true distribution in Table 1. For details see Appendix 1.3. Indeed, one could use estimates that include the covariate instead of the Kaplan-Meier estimate, so as to estimate $\mathbb{P}(C > t|x)$. Since this requires a regression of the censoring distribution to compute a score of the regression of the event-time, this raises the question of how to choose the best regression of the censoring distribution. Because of this loop, in practice the Kaplan-Meier estimate is often chosen (Kvamme and Borgan, 2019; Kvamme et al., 2019). The same analysis of the Brier score can be applied to the binomial log-likelihood (BLL). In particular, the BLL is a proper score under the assumption that $C \perp\!\!\!\perp X$, but it need not be proper when this assumption is violated. We use Example 2 to demonstrate that BLL also breaks in Table 1. It should be stressed that the pathology of Brier Score lies in the difficulty of estimating the weights rather than the score itself; work such as Han et al. (2021), propose inverse weighted survival games to correctly estimate the Brier Score.

Integrated Brier score and integrated binomial log-likelihood The Brier score above was defined for a fixed time t . The integrated Brier score (IBS) has been defined as a measure of goodness-of-fit

of the estimate $\hat{S}(\cdot)$ by

$$\begin{aligned} \mathcal{S}_{\text{IBS}}(\hat{S}(t|x), (z, d)) \\ = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \mathcal{S}_{\text{Brier}}^t(\hat{S}(t|x), (z, d)) dt, \end{aligned}$$

where typically $t_1 = 0$ and $t_2 = z_{\max} = \max\{z_1, \dots, z_n\}$. In experiments, the integral is approximated by a sum over an equally spaced grid of times. The binomial log-likelihood at time t , $\mathcal{S}_{\text{BLL}}^t$ is defined by:

$$\begin{aligned} \mathcal{S}_{\text{BLL}}^t(S(t|x), (z, d)) = \\ \frac{\log(1 - \hat{S}(t|x))1\{z \leq t, d = 1\}}{\hat{G}(z)} + \frac{S(t|x)1\{z \geq t\}}{\hat{G}(t)}. \end{aligned}$$

Analogously to the definition of IBS, the integrated binomial log-likelihood, IBLL, is defined as

$$\mathcal{S}_{\text{IBLL}}(S(t|x), (z, d)) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \mathcal{S}_{\text{BLL}}^t(S(t|x), (z, d)) dt.$$

As neither $\mathcal{S}_{\text{Brier}}^t$ nor $\mathcal{S}_{\text{BLL}}^t$ are proper scores, it follows that $\mathcal{S}_{\text{Integrated-Brier}}$ and $\mathcal{S}_{\text{IBLL}}$ are not proper scores. We use Example 2 to show that IBS and IBLL breaks in Table 1.

Survival-CRPS The survival continuous ranked probability score (survival-CRPS) is defined by

$$\begin{aligned} \mathcal{S}_{\text{CRPS}}(\hat{F}_X, (Z, D)) = \\ \int_0^Z \hat{F}(t|X)^2 dt + D \int_Z^\infty (1 - \hat{F}(t|X))^2 dt \end{aligned}$$

and was proposed in Avati et al. (2020) as an extension of the continuous ranked probability score (CRPS) (Gneiting et al., 2007). This is equivalent to the integrated Brier score without inverse probability weighting. In Avati et al. (2020), the score was also claimed to be a proper score. The proof strategy given there argues that the survival-CRPS is simply the weighted CRPS of Gneiting and Ranjan (2011) with the weight function set to the indicator of uncensored regions. As we show in Appendix 1.4.1, however, this weight function does not recover the survival-CRPS. This suggests that the survival-CRPS is not a proper score. For example, if the censoring time C is deterministic, i.e., $C = c$ w.p. 1 for $c \in \mathbb{R}_{\geq 0}$, then it is clear that to minimize the survival-CRPS, one would set $\hat{F}(c) = 1$, even if the true CDF is less than 1. This principle also holds in the case of random censoring, where one can construct false CDFs which yield a lower (i.e., better) survival-CRPS. We use this insight to construct an example (Example 3) where survival-CRPS breaks in Table 1. For details, see Appendix 1.4.2.

3.2 The right-censored log-likelihood is a proper score

The right-censored log-likelihood score of an observation (z, d) , given a covariate x and distribution $S(t|x)$ is simply the likelihood given in Equation 1, i.e.:

$$\mathcal{S}_L(S(t|x), (z, d)) = d \log f(z|x) + (1 - d) \log S(z|x).$$

Let \hat{S} denote any continuous survival curve, and let S denote the true survival distribution of the event-time T , and make the following assumption.

Assumption 2. Assume that $KL(S||\hat{S}) < \infty$, where KL denotes the Kullback–Leibler divergence.

Under this assumption, we prove the following lemma.

Lemma 1. Let S be the true survival distribution of the event-time T . For every distribution \hat{S} that satisfies Assumption 2 and for every $x \in \mathbb{R}^p$, it holds that

$$\begin{aligned} \mathbb{E}[\mathcal{S}_L(S(t|x), (Z, D))|X = x] \\ \geq \mathbb{E}[\mathcal{S}_L(\hat{S}(t|x), (Z, D))|X = x] \end{aligned}$$

Proof. We need to prove that for every $x \in \mathbb{R}^p$, for every distribution $S(t|x)$

$$\begin{aligned} \mathbb{E}[\mathcal{S}(S(t|x), (Z, D))|X = x] \\ \geq \mathbb{E}[\mathcal{S}(\hat{S}(t|x), (Z, D))|X = x] \end{aligned}$$

where $T \sim S(t|x)$, the censoring time C follows an arbitrary distribution and it is assumed that $S(t|x)$ and $\hat{S}(t|x)$ satisfy Assumption 1. We can safely omit x from the notation and prove that

$$\mathbb{E}\mathcal{S}(S, (Z, D)) \geq \mathbb{E}\mathcal{S}(\hat{S}, (Z, D))$$

where again $T \sim S$, the censoring time C follows an arbitrary distribution and it is assumed that S and \hat{S} satisfy Assumption 1. This has been proved in a slightly different context in Diks et al. (2011) Lemma 1. In that context the censoring variable is assumed known. The same argument can be applied to the case of random censoring, by first conditioning on the censoring variable. Namely, if we can prove that

$$\mathbb{E}[\mathcal{S}(S, (Z, D))|C] \geq \mathbb{E}[\mathcal{S}(\hat{S}, (Z, D))|C]$$

then it follows, by taking the expectation over C , that also

$$\mathbb{E}\mathcal{S}(S, (Z, D)) \geq \mathbb{E}\mathcal{S}(\hat{S}, (Z, D)).$$

We now prove the first inequality following the proof of Lemma 1 in Diks et al. (2011). Let f denote the density of S , and let \hat{f} denote the density of \hat{S} . Note that the expected score can be written as follows:

$$\mathbb{E}[\mathcal{S}(\hat{S}, (Z, D))|C] =$$

$$\begin{aligned}
 & \mathbb{E} \left[D \log \hat{f}(Z) + (1 - D) \log \hat{S}(Z) \mid C \right] \\
 &= \int_0^\infty 1\{t \leq C\} f(t) \log \hat{f}(t) + 1\{t > C\} f(t) \log \hat{S}(C) dt \\
 &= \int_0^C f(t) \log \hat{f}(t) dt + S(C) \log \hat{S}(C).
 \end{aligned}$$

Hence,

$$\begin{aligned}
 & \mathbb{E}[\mathcal{S}(S, (Z, D)) - \mathcal{S}(\hat{S}, (Z, D)) \mid C] \\
 &= \int_0^C f(t) \log \frac{f(t)}{\hat{f}(t)} dt + S(C) \log \frac{S(C)}{\hat{S}(C)} \\
 &= F(C) \int_0^C \frac{f(t)}{F(C)} \log \frac{f(t)/F(C)}{\hat{f}(t)/\hat{F}(C)} dt + S(C) \log \frac{S(C)}{\hat{S}(C)} - \\
 & F(C) \int_0^C \frac{f(t)}{F(C)} \log \frac{1/F(C)}{1/\hat{F}(C)} dt \\
 &= F(C) \int_0^C \frac{f(t)}{F(C)} \log \frac{f(t)/F(C)}{\hat{f}(t)/\hat{F}(C)} dt + \\
 & S(C) \log \frac{S(C)}{\hat{S}(C)} + F(C) \log \frac{F(C)}{\hat{F}(C)} \\
 &= F(C) \text{KL}(f(t)/F(C) \parallel \hat{f}(t)/\hat{F}(C)) + \\
 & \text{KL}(\text{Ber}(S(C)) \parallel \text{Ber}(\hat{S}(C))) \geq 0.
 \end{aligned}$$

The distributions $f(t)/F(C)$ and $\hat{f}(t)/\hat{F}(C)$ are the densities of f and \hat{f} conditioned on the event $T \leq C$. The first KL divergence exists by Assumption 1. The second term is the KL divergence between Bernoulli random variables with success probabilities $S(C)$ and $\hat{S}(C)$ respectively. \square

Hence, under Assumption 2, the right-censored log-likelihood is a proper scoring rule. We provide a proof in Appendix 1.1, as a simple extension of Lemma 1 of Diks et al. (2011), by conditioning on the censoring variable and the covariate. We note that this result may also be deduced from Section 3.5 of Dawid and Musio (2014), as a non-trivial special case where one uses the function $\psi(p) = p \log p - p$, but the more general claim there is provided without a proof. We empirically validate that the right-censored log-likelihood is a proper score by applying it to the true and fake distributions for Example 1, 2 and 3 in Table 1. As expected, the right-censored log-likelihood correctly gives the true distribution a higher score than the fake distribution for all examples.

Summary of insights We have shown both theoretically and empirically that $\mathcal{S}_C^{\text{td}}$, \mathcal{S}_{IBS} , $\mathcal{S}_{\text{IBLL}}$ and $\mathcal{S}_{\text{CRPS}}$ are not proper scores, meaning that optimizing them gives no guarantee of learning the true survival distribution for right-censored data. Since the right-censored log-likelihood is proven to be a proper score,

it is imperative to construct models that train and evaluate on the log-likelihood directly, rather than using non-proper proxy scores.

4 OPTIMIZING LIKELIHOOD USING PARTIALLY MONOTONIC NEURAL NETWORKS

To optimize the right-censored log-likelihood of (1), one needs to jointly model the survival distribution and the density. To do so, we adapt the monotonic neural density estimator (MONDE) of Chilinski and Silva (2020) in the case of a univariate response variable to the survival context. A similar strategy has been used in Omi et al. (2019) to model temporal point processes. We adapt MONDE to jointly model the survival function and the density function of a right-censored event-time, resulting in a scalable and flexible survival regression model that can train and evaluate directly on log-likelihood. The model is coined *Survival Monotonic-neural network* (SuMo-net) and we provide a brief overview of its architecture in Figure 2.

Architecture Figure 2 pictures the SuMo-network structure as a directed graph with two types of edges. The network can be defined in terms of two fully connected feed-forward sub-networks. A covariate x is first passed through a network $h^{\text{cov}}(x)$ with L^{cov} layers in which each layer is fully connected to the next, and no other edges are present. The output of the covariate network, together with the time input t forms the input of a second network $h^{\text{mixed}}(t, u)$, where u has the same dimension as the output of the covariate network $h^{\text{cov}}(x)$. In this network, too, all edges are present from layer l to $l + 1$ for $l = 1, \dots, L^{\text{mixed}} - 1$ and no additional edges are added. The output layer of the mixed network consists of a single neuron with value $h(t, x)$, thus returning a scalar. The total network h thus equals

$$h(t, x) = h^{\text{mixed}}(t, h^{\text{cov}}(x)).$$

We then model $S(t|x)$ through $\hat{S}(t|x) = 1 - \text{sigmoid}(h(t, x))$. It should be noted that we propose a minimal implementation of SuMo-net, only using a vanilla feed-forward architecture. Any other type of architecture such as Resnet (He et al., 2015), LSTM (Hochreiter and Schmidhuber, 1997), Densenet (Huang et al., 2018) etc., can be used to enhance model flexibility further, depending on the nature of the application.

Training After passing $h(t, x)$ through the sigmoid function, the final edge in Figure 2 represents differentiation of $\text{sigmoid}(h(t, x))$ with respect to t and we set $\hat{f}(t|x) = -\partial \hat{S}(t|x) / \partial t$. To compute the derivative, we use automatic differentiation as implemented in PyTorch (Paszke et al., 2019), in line with the approach

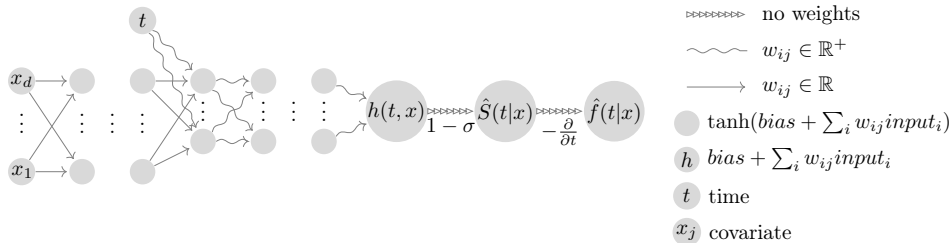


Figure 2: [Figure adapted from Figure 1 of Chilinski and Silva (2020)] The graph representing SuMo-net, modelling the survival curve. The last edge symbolizes the operation of differentiating the conditional survival distribution function $\hat{S}(t|x)$ with respect to the input t , resulting in the conditional density of the survival time. The second-to-last edge represents passing $h(t, x)$ through $1 - \text{sigmoid}$. If one models the cumulative hazard instead, we replace $1 - \text{sigmoid}$ by SoftReLU and use $\frac{\partial}{\partial t}$ in the last edge instead of $-\frac{\partial}{\partial t}$.

taken in Chilinski and Silva (2020). One can also approximate the derivative by applying the chain rule to $\text{sigmoid}(h(t, x))$, yielding that

$$f(t|x) = -\text{sigmoid}(h(t, x))(1 - \text{sigmoid}(h(t, x))) \times \left(\frac{h(t + \epsilon, x) - h(t, x)}{\epsilon} \right) + o(1).$$

We implemented SuMo-net both using this approximation and using auto-grad and found the implementations performed comparably during the experiments. We found further that both of these implementations improved performance over simply evaluating $-(S(t + \epsilon|x) - S(t|x))/\epsilon$, which may be due to the vanishing gradient of the sigmoid function as one moves away from 0.

Universality Since the survival function and the cumulative hazard function are non-increasing and non-decreasing in t respectively, we can ensure that the output of the neural net does not decrease as t increases by restricting the weights which are applied to the time t to be non-negative. As discussed in Lang (2005), non-decreasing behaviour in t is guaranteed if all weights that are descendants of t in Figure 2 are non-negative. In Lang (2005) it is furthermore shown that if there are at least 2 hidden layers in the mixed network $h_{\text{mixed}}(t, u)$, then, given enough nodes in each layer, the network can universally approximate any true survival function. In our implementation, we ensure that weights are non-negative by defining them as the square of unrestricted weights. It should be noted that despite being a relatively simple idea, the neural network property of SuMo-net allows survival regression to straightforwardly use *any type of data* such as images, graphs or any type of sequences, which allows for an unrivaled flexibility in data choices.

5 EXPERIMENTS

We apply SuMo-net to five real datasets SUPPORT, METABRIC, GBSG, FLCHAIN, and KKBOX presented in Kvamme et al. (2019). We compare the

results with the methods DeepHit (Lee et al., 2018), Cox-Time and Cox-CC (Kvamme et al., 2019), DeepSurv (Katzman et al., 2018) and the classical Cox model (Cox, 1972). To nuance the comparisons, we contrast against reported results in SODEN Tang et al. (2020) and SurvNODE Groha et al. (2021), which are both NeuralODE based models that can train on likelihood directly. We also compare against modelling the parameters (scale & shape and mean & variance) of the Weibull and Log-normal distribution with neural nets as functions of covariates and time, analogous to Avati et al. (2020). When parametrising a survival distribution directly, the right-censored likelihood can in fact be calculated directly.

We use a 5-fold cross-validation procedure in which the model is trained on three folds, one fold is used for validation, and one fold is reserved for testing. We then average the evaluation criteria calculated on five different test folds. For datasets FLCHAIN, GBSG, METABRIC, and SUPPORT we optimize SuMo-net and baselines by using the hyperopt library (Bergstra et al., 2013) for 300 iterations. For the KKBOX dataset we do the same, but for 100 hyperopt iterations, due to computational constraints. All experiments were run on a GPU cluster consisting of 8 Nvidia 1080 GTX TI cards.

Results In Table 3 we compare the methods in terms of right-censored likelihood. As the benchmarks do not have tractable right-censored likelihood, this is approximated by interpolating their survival curves. While the obtained likelihoods of the benchmarks are only approximations, the results suggest SuMo-net is performant in terms of right-censored likelihood even when compared to much more complex NeuralODE based methods. It should be noted that neither SurvNODE or SODEN has been run on datasets larger than 40000 observations, due to computational costs scaling with observations and function evaluations (Massaroli et al., 2021). In Table 4 we compare the binomial log-likelihood of the models. Also in terms of IBLL, SuMo-net scores well. Results for IBS and time-dependent concordance are

Dataset	SUPPORT	METABRIC	GBSG	FLCHAIN	KKBOX	Weibull	Checkerboard	Normal
Size	8,873	1,904	2,232	6,524	2,646,746	25,000	25,000	25,000
Covariates	14	9	7	8	15	1	1	1
Unique Durations	1,714	1,686	1,230	2,715	2,344,004	25000	25000	25000
Prop. Censored	0.32	0.42	0.43	0.70	0.28	0.55	0.75	0.50

Table 2: Summary of all datasets used in the experiments. Weibull, checkerboard and normal are synthetic datasets and the others are real datasets. See Kvamme et al. (2019) for more details on the real datasets.

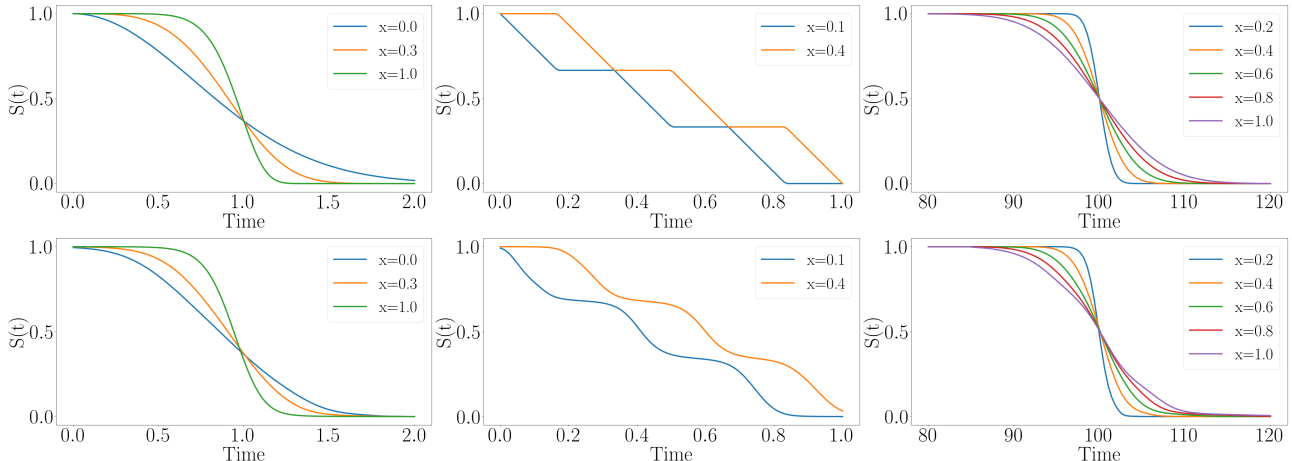


Figure 3: True distributions (top row) and learned distributions using SuMo-net (bottom row). From left to right, the distributions are Weibull, Checkboard, and Normal.

given in the Appendix. Notably, we find that DeepHit, which optimizes a concordance-like loss function, generally has the best performance in terms of time-dependent concordance, yet achieves the worst scores in terms of likelihood. This agrees with our findings that time-dependent concordance is not a proper scoring rule and that optimizing time-dependent concordance may lead to inaccurate distributions.

Correlation between likelihood and existing metrics Following our result in Section 3 that likelihood is the only proper score among those considered, we investigate how the other scores compare to likelihood by calculating the rank correlation between all scores. In particular, for each of the datasets FLCHAIN, GBSG, METABRIC, SUPPORT and KKBOX and each hyperparameter run of SuMo-net, we compute all scores on the test set and turn them into ranks. The correlation matrix averaged over the datasets is displayed in Table 5. The experiment indicates that concordance is almost independent of likelihood, while IBLI exhibits the strongest correlation, suggesting that it could serve as a good proxy when the likelihood is not available.

Estimation of the likelihood for models other than SuMo-net All the baseline models proposed in Kvamme et al. (2019), learn discrete distributions and hence have no density and the likelihood cannot be computed. For those methods we use interpolation of the survival curve to approximate the density. Let the survival curve of the discrete distribution be denoted

by S and denote the points at which the survival curve makes jumps by $t_0 \leq \dots \leq t_m$. Then, given a time t , we find i so that $t_i \leq z < t_{i+1}$ and then estimate

$$\hat{f}(t) = -\frac{S(t_{i+k}) - S(t_{i-k+1})}{t_{i+k} - t_{i-k+1}}, \quad (4)$$

where k determines the width of the interval we use to approximate the derivative. We found that setting $k = 1$ resulted in a highly variable density and low likelihood, and hence we use $k = 2$ in the main text. SuMo-net has tractable likelihood, so this approximation does not have to be made. While comparing models with an approximated and exact likelihood score is not ideal, it illustrates a general issue with modern survival models: they are not endowed with a tractable likelihood, making it difficult to evaluate them on a proper score at test time. As likelihood is an easy proper score to calculate for survival models, we stress that future models should be developed with tractable likelihood evaluation procedure in mind.

Calibration When training survival models on likelihood directly we have direct access to the density $f(t|x)$ and to the cumulative distribution $F(t|x)$, which means we can quantify how calibrated the fitted survival distribution is. Following Kuleshov et al. (2018) and Goldstein et al. (2021), a calibrated distribution would have an empirical frequency $\hat{p}_j = \frac{|\{z_i | F(z_i|x_i) \leq p_j, i=1, \dots, n\}|}{n}$ equal to p_j . The calibra-

	CoxCC	Cox-Linear	Cox-Time	DeepHit	$S_L \uparrow$ DeepSurv	SODEN	Weibull	Log-normal	SuMo-net
FLCHAIN	-0.507 ± 0.014	-0.513 ± 0.014	-0.584 ± 0.239	-0.85 ± 0.493	-0.529 ± 0.015	N/A	-0.467 ± 0.085	-0.535 ± 0.251	-0.376 ± 0.008
GBSG	-0.501 ± 0.038	-0.498 ± 0.04	-0.454 ± 0.034	-0.719 ± 0.19	-0.502 ± 0.04	N/A	-0.577 ± 0.117	-0.563 ± 0.115	-0.391 ± 0.027
KKBOX	0.797 ± 0.013	0.706 ± 0.01	1.411 ± 0.014	-4.174 ± 4.128	0.758 ± 0.036	N/A	-0.118 ± 0.0	0.201 ± 0.289	1.553 ± 0.132
METABRIC	-0.305 ± 0.037	-0.316 ± 0.031	-0.233 ± 0.066	-0.565 ± 0.348	-0.318 ± 0.029	-0.149 ± 0.015	-0.29 ± 0.042	-0.299 ± 0.051	-0.142 ± 0.042
SUPPORT	0.533 ± 0.047	0.545 ± 0.03	0.57 ± 0.038	0.098 ± 0.141	0.549 ± 0.045	-0.676 ± 0.009	0.035 ± 0.024	0.527 ± 0.098	0.658 ± 0.027

Table 3: Log-likelihood scores and standard deviations over the 5 folds. Higher is better. SuMo-net, Weibull, Log-normal and SODEN provides exact likelihood. For other methods we have approximated the likelihood. In particular, we approximate the density by interpolating the survival curve. See Equation (4) for details.

	CoxCC	Cox-Linear	Cox-Time	DeepHit	$S_{IBLL} \uparrow$ DeepSurv	SurvNODE	SODEN	Weibull	Log-normal	SuMo-net
FLCHAIN	-0.464 ± 0.194	-0.336 ± 0.012	-0.47 ± 0.241	-0.646 ± 0.462	-0.339 ± 0.01	N/A	N/A	-0.389 ± 0.045	-0.504 ± 0.209	-0.332 ± 0.006
GBSG	-0.546 ± 0.02	-0.549 ± 0.011	-0.541 ± 0.021	-1.157 ± 1.287	-0.539 ± 0.013	N/A	N/A	-1.513 ± 1.525	-1.093 ± 0.633	-0.528 ± 0.012
KKBOX	-0.382 ± 0.005	-0.429 ± 0.001	-0.33 ± 0.004	-0.745 ± 0.513	-0.368 ± 0.002	N/A	N/A	-3.032 ± 1.766	-1.776 ± 2.187	-0.395 ± 0.029
METABRIC	-0.53 ± 0.035	-0.522 ± 0.012	-0.526 ± 0.012	-0.656 ± 0.156	-0.52 ± 0.013	-0.477	-0.484 ± 0.010	-0.554 ± 0.004	-0.597 ± 0.064	-0.485 ± 0.015
SUPPORT	-0.572 ± 0.008	-0.577 ± 0.005	-0.577 ± 0.009	-0.701 ± 0.065	-0.569 ± 0.01	-0.580	-0.561 ± 0.002	-0.705 ± 0.013	-0.654 ± 0.053	-0.569 ± 0.004

Table 4: IBLL scores and standard deviations over the 5 folds. Higher is better

	S_L	$S_{C^{td}}$	S_{IBS}	S_{IBLL}
S_L	1.000	0.174	0.688	0.851
$S_{C^{td}}$	0.174	1.000	0.223	0.179
S_{IBS}	0.688	0.223	1.000	0.841
S_{IBLL}	0.851	0.179	0.841	1.000

Table 5: Rank correlation matrix between likelihood, concordance, IBS and IBLL. The correlation matrix is averaged over all datasets.

	Calibration score		
	Weibull	Log-normal	SuMo-net
FLCHAIN	1.846 ± 0.641	0.934 ± 0.579	1.165 ± 0.116
GBSG	0.762 ± 0.442	0.635 ± 0.245	0.128 ± 0.03
KKBOX	0.897 ± 0.401	0.596 ± 0.443	0.112 ± 0.077
METABRIC	0.327 ± 0.177	0.561 ± 0.517	0.097 ± 0.056
SUPPORT	0.342 ± 0.276	0.237 ± 0.139	0.313 ± 0.049

Table 6: SuMo-net generally provides a calibrated survival distribution

tion error is then defined as $\text{cal}(F_1, y_1, \dots, F_T, y_T) = \sum_{j=1}^m w_j \cdot (p_j - \hat{p}_j)^2$, where we take $w_j = 1, \forall j$ and $p_j \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. As the empirical frequency can only be coarsely approximated without direct access to $F(t|x)$, we only evaluate calibration for methods that train directly on likelihood.

Inference and training timings Here we contrast the timing of predictions of the survival probability $S(t|x)$ using a trained model. We compare the time of SuMo-net against the time of Cox-Time presented in Kvamme et al. (2019) and SODEN (Tang et al., 2020). To compute the survival probability, Cox-Time integrates the learned hazard function over time, which is computationally costly. In contrast, SuMo-net evaluates S through a single forward pass. To compare the timings we initialize SuMo-net, Cox-Time and SODEN using an identical architecture and compare inference timings on the test-fold of all datasets in Table 7. The findings indicate SuMo-net does indeed achieve its anticipated speedup.

	SODEN (s)	Cox-Time (s)	SuMo-net (s)	Speedup vs	
				SODEN	Cox-Time
SUPPORT	1.124 ± 0.070	2.97 ± 0.126	0.047 ± 0.004	24	63
METABRIC	0.459 ± 0.044	1.573 ± 0.126	0.016 ± 0.001	29	100
GBSG	N/A	1.46 ± 0.112	0.021 ± 0.002	N/A	69
FLCHAIN	N/A	4.199 ± 0.246	0.044 ± 0.003	N/A	96
KKBOX	N/A	140.505 ± 7.976	7.034 ± 1.092	N/A	20

Table 7: Inference timings on 20% of the data. SuMo-net achieves a speedup factor of between 20 and 100.

6 CONCLUSION

We have shown that several existing scoring rules for right-censored regression are not proper, i.e., they may not faithfully reflect the accuracy of a fitted distribution. We also proved that the right-censored log-likelihood is a proper scoring rule. While other scoring rules can still be useful for certain purposes, their limitations should be taken into consideration. Since the right-censored log-likelihood is thus attractive both as a scoring rule and to assess model fit, we proposed SuMo-net, a simple, scalable and flexible regression model for survival data, for which the right-censored log-likelihood is tractable by construction. Future fruitful directions would be to extend SuMo-net to more involved architectures and datatypes such as genetic data, image data and other high dimensional data.

Downstream applications While SuMo-net considers a monotonic neural network, the idea of enforcing monotonicity on the time-to-event variable t can be extended to any model with trainable weights to train directly on any type of censored survival likelihood. This allows fields such as epidemiology, medical statistics, and economics, among many others, to formulate highly flexible non-parametric models that can evaluate directly on a proper scoring rule instead of non-proper proxy objectives. This is particularly important in safety critical domains where accurately approximating the true underlying model impacts subsequent decision making. We hope that the introduction of SuMo-net can inspire future developments of sophisticated survival models endowed with a tractable likelihood for proper evaluation and comparison between models.

ACKNOWLEDGMENTS

We thank the reviewers for their helpful remarks. We are grateful for our colleagues Jean-Francois Ton, Rob Zinkov, Siu Lun Chau and Zoi Tsangalidou for their helpful comments and remarks.

References

- Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.
- Anand Avati, Tony Duan, Sharon Zhou, Kenneth Jung, Nigam H Shah, and Andrew Y Ng. Countdown regression: sharp and calibrated survival predictions. In *Uncertainty in Artificial Intelligence*, pages 145–155. PMLR, 2020.
- J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML’13*, page I–115–I–123. JMLR.org, 2013.
- Paul Blanche, Michael W Kattan, and Thomas A Gerds. The c-index is not proper for the evaluation of t -year predicted risks. *Biostatistics*, 20(2):347–357, 02 2018. ISSN 1465-4644. doi: 10.1093/biostatistics/kxy006. URL <https://doi.org/10.1093/biostatistics/kxy006>.
- Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Courtney Page, Benjamin Goldstein, Lawrence Carin Duke, and Ricardo Henao. Adversarial time-to-event modeling. In *International Conference on Machine Learning*, pages 735–744. PMLR, 2018.
- Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. In *Conference on Uncertainty in Artificial Intelligence*, pages 420–429. PMLR, 2020.
- David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.
- Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *Metron*, 72(2):169–183, 2014.
- Cees Diks, Valentyn Panchenko, and Dick Van Dijk. Likelihood-based scoring rules for comparing density forecasts in tails. *Journal of Econometrics*, 163(2): 215–230, 2011.
- Lore Dirick, Gerda Claeskens, and Bart Baesens. Time to default in credit scoring using survival analysis: a benchmark study. *Journal of the Operational Research Society*, 68(6):652–665, 2017.
- Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477): 359–378, 2007.
- Tilmann Gneiting and Roopesh Ranjan. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, 29(3):411–422, 2011.
- Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.
- Mark Goldstein, Xintian Han, Aahlad Puli, Adler J. Perotte, and Rajesh Ranganath. X-cal: Explicit calibration for survival analysis, 2021.
- Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17-18):2529–2545, 1999.
- Stefan Groha, Sebastian M Schmon, and Alexander Gusev. A general framework for survival analysis and multi-state modelling, 2021.
- Chong Gu. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media, 2013.
- Xintian Han, Mark Goldstein, Aahlad Puli, Thomas Wies, Adler J Perotte, and Rajesh Ranganath. Inverse-weighted survival games, 2021.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.
- Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.
- Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008.

- Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):24, 2018.
- David G. Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text*. Springer Science and Business Media, LLC, New, 2005.
- Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression, 2018.
- Håvard Kvamme and Ørnulf Borgan. The brier score under administrative censoring: Problems and solutions. *arXiv preprint arXiv:1912.08581*, 2019.
- Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and Cox regression. *Journal of machine learning research*, 20(129):1–30, 2019.
- Bernhard Lang. Monotonic multi-layer perceptron networks as universal approximators. In *International conference on artificial neural networks*, pages 31–37. Springer, 2005.
- John A Laurie, Charles G Moertel, Thomas R Fleming, Harry S Wieand, John E Leigh, Jebal Rubin, Greg W McCormack, James B Gerstner, James E Krook, and James Malliard. Surgical adjuvant therapy of large-bowel carcinoma: an evaluation of levamisole and the combination of levamisole and fluorouracil. the north central cancer treatment group and the Mayo Clinic. *Journal of Clinical Oncology*, 7(10):1447–1456, 1989.
- Changhee Lee, William R Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *AAAI*, pages 2314–2321, 2018.
- Gilbert MacKenzie. The statistical analysis of failure time data. *The Statistician*, 31:278–278, 1982.
- Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes, 2021.
- James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.
- Xenia Miscouridou, Adler Perotte, Noémie Elhadad, and Rajesh Ranganath. Deep survival analysis: Non-parametrics and missingness. In *Machine Learning for Healthcare Conference*, pages 244–256. PMLR, 2018.
- Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. *Fully Neural Network Based Model for General Temporal Point Processes*. Curran Associates Inc., Red Hook, NY, USA, 2019.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning for Healthcare Conference*, pages 101–114. PMLR, 2016.
- Dimitris Rizopoulos, Laura A Hatfield, Bradley P Carlin, and Johanna JM Takkenberg. Combining dynamic predictions from joint models for longitudinal and time-to-event data using bayesian model averaging. *Journal of the American Statistical Association*, 109(508):1385–1397, 2014.
- Gian A. Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2015.
- Weijing Tang, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu. Soden: A scalable continuous-time survival model through ordinary differential equation networks, 2020.
- Dirk Van den Poel and Bart Larivière. Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 157(1):196–217, 2004.
- Matthias Ziehm and Janet Thornton. Unlocking the potential of survival data for model organisms through a new database and online analysis platform: Survcurv. *Aging cell*, 12, 07 2013. doi:10.1111/accel.12121.

Supplementary Material: Survival Regression with Proper Scoring Rules and Monotonic Neural Networks

A Appendix

A.1 Details on time-dependent concordance example

Let $X \in \{0, 1\}$. Let $K \sim \text{Uniform}\{0, 2, 4, 6, 8\}$ and let $U \sim \text{Uniform}[0, 1]$. Then $T|X = 0 \sim K + U$ and $T|X = 1 \sim K + U + 1$. We further set $C \sim \text{Exponential}(\text{mean} = 20)$. Note that the hazard rates of the two distributions are nonzero on disjoint sets. The False distribution in Figure 1, is constructed such that at each time, the distribution with the nonzero hazard rate has the lowest survival probability.

A.2 Details on Brier score example

Let $X \sim \text{Bernoulli}(0.1)$. We create an extreme example of unequal censoring in which $C \sim \text{Exp}(1)$ if $X = 0$ and $C = \infty$ if $X = 1$. Let $T \sim \text{Exponential}(10)$ (independent of X), so that $F_{\text{True}}(t) = 1 - \exp(-t/10)$. Finally create a fake distribution \hat{F} so that $\hat{F}(t|X = 1) = F_{\text{true}}(t)$ and $\hat{F}(t|X = 0) = 1 - \exp(-t/4)$. We take a sample of size $n = 10^6$. We evaluate the Brier score at $t = 4$. We evaluate the Integrated Brier score from $t = 0$ to $t = 40$. We repeat the procedure for BLL and IBLL.

A.3 Discussion of survival-CRPS

A.3.1 Discussion of proof given in Avati et al

In Gneiting and Ranjan (2011) it is stated that the following score is proper (see Matheson and Winkler (1976) for a proof),

$$\begin{aligned} S(F, z) &= \int_0^\infty (F(r) - 1\{r \geq z\})^2 u(r) dr \\ &= \int_0^z F(r)^2 u(r) dz + \int_z^\infty (1 - F(r))^2 u(r) dr. \end{aligned}$$

for non-negative weight functions u . In Avati et al. (2020) it is argued that the survival-CRPS is simply the above score, with weight function equal to the uncensored region. It is not immediately clear how this is meant to be interpreted, as the weight function needs to be a non-random function. If it were a proper score when conditioned on a known censoring time $C = c$, for every value of c , then we could proceed as in Section 3.2, using the conditional independence of C and T . Assuming $C = c$, the uncensored region equals $u(r) = 1\{r \leq c\}$. Proceeding with this as the weight function we would obtain

$$\begin{aligned} S(F, z) &= \int_0^\infty (F(r) - 1\{r \geq z\})^2 1\{r \leq c\} dr \\ &= \int_0^z F(r)^2 1\{r \leq c\} dz + \int_z^\infty (1 - F(r))^2 1\{r \leq c\} dr. \end{aligned}$$

Note that $z \leq c$ since $z = \min(t, c)$ and therefore the first indicator is always true and can be removed, to arrive at:

$$\begin{aligned}
S(F, z) &= \int_0^z F(r)^2 dr + \int_z^\infty (1 - F(r))^2 \mathbf{1}\{r \leq c\} dr \\
&= \int_0^z F(r)^2 dr + \int_z^c (1 - F(r))^2 dr.
\end{aligned}$$

Hence, the survival-CRPS has an additional term

$$d \int_c^\infty (1 - F(r))^2 dr \quad (5)$$

which is missing from the above expression. This shows, at least, that this interpretation of the proof strategy would not succeed. Furthermore, assuming that c is a fixed censoring time, then $F(r)$ is only evaluated for $r > c$ in Equation 5. Clearly to minimize the survival-CRPS, one then sets $F(r) = 1$ for $r \geq c$, even if the true distribution satisfies $F_{\text{true}}(r) < 1$ for $r \geq c$.

A.3.2 An example

Assume for simplicity that there are no covariates. Let $T \sim \text{Exp}(\text{mean} = 100)$ with every individual set to fail at $t = 200$ so that $F_{\text{True}}(t) = 1 - \exp(-t/100)$ for $t < 200$ and $F_{\text{true}}(200) = 1$. Let $C \sim \text{Exp}(\text{mean} = 10)$. Observations are thus likely to be censored. Let a fake distribution be $F_{\text{False}} = \text{Exp}(\text{mean} = 25)$. We then take a sample of size 10^6 from the fake distribution and true distribution and evaluate and compute their survival-CRPS against the true distribution.

A.4 Calculating life expectancies using SuMo-net

The life expectancies are calculated by direct numerical integration using the learned survival distribution $S_\theta(t|x)$ since $\mathbb{E}[T|x] = \int_0^\infty S_\theta(t|x) dt$.

A.5 Log-likelihood scores for $d = 1, 4, 8$

dataset Method	$\mathcal{S}_L \uparrow$				
	FLCHAIN	GBSG	KKBOX	METABRIC	SUPPORT
CoxCC	-0.583 ± 0.014	-0.632 ± 0.032	0.156 ± 0.075	-0.451 ± 0.06	0.437 ± 0.06
Cox-Linear	-0.586 ± 0.026	-0.603 ± 0.032	0.05 ± 0.111	-0.444 ± 0.03	0.451 ± 0.053
Cox-Time	-0.565 ± 0.071	-0.569 ± 0.063	0.926 ± 0.09	-0.346 ± 0.056	0.481 ± 0.047
DeepHit	-0.878 ± 0.496	-0.805 ± 0.212	-1.432 ± 0.226	-0.686 ± 0.5	0.04 ± 0.159
DeepSurv	-0.605 ± 0.01	-0.609 ± 0.04	0.179 ± 0.073	-0.455 ± 0.027	0.446 ± 0.036
SuMo-net	-0.376 ± 0.008	-0.391 ± 0.027	1.553 ± 0.132	-0.142 ± 0.042	0.658 ± 0.027

Table 8: Log-likelihood scores for $d = 1$

Dataset Method	$\mathcal{S}_L \uparrow$				
	FLCHAIN	GBSG	KKBOX	METABRIC	SUPPORT
CoxCC	-0.482 ± 0.01	-0.451 ± 0.033	0.181 ± 0.088	-0.286 ± 0.054	0.564 ± 0.04
Cox-Linear	-0.49 ± 0.017	-0.465 ± 0.043	0.095 ± 0.113	-0.265 ± 0.02	0.565 ± 0.032
Cox-Time	-0.496 ± 0.166	-0.409 ± 0.043	0.93 ± 0.089	-0.189 ± 0.061	0.6 ± 0.041
DeepHit	-0.846 ± 0.497	-0.94 ± 0.415	-4.454 ± 4.441	-0.56 ± 0.338	0.113 ± 0.126
DeepSurv	-0.498 ± 0.015	-0.457 ± 0.021	0.158 ± 0.117	-0.282 ± 0.027	0.562 ± 0.029
SuMo-net	-0.376 ± 0.008	-0.391 ± 0.027	1.553 ± 0.132	-0.142 ± 0.042	0.658 ± 0.027

Table 9: Log-likelihood scores for $d = 4$

Survival Regression with Proper Scoring Rules and Monotonic Neural Networks

Dataset Method	$\mathcal{S}_L \uparrow$				
	FLCHAIN	GBSG	KKBOX	METABRIC	SUPPORT
CoxCC	-0.485 ± 0.014	-0.451 ± 0.033	0.42 ± 0.021	-0.243 ± 0.023	0.568 ± 0.036
Cox-Linear	-0.477 ± 0.012	-0.442 ± 0.033	0.344 ± 0.037	-0.253 ± 0.012	0.572 ± 0.037
Cox-Time	-0.55 ± 0.178	-0.398 ± 0.036	0.985 ± 0.037	-0.179 ± 0.064	0.599 ± 0.043
DeepHit	-0.835 ± 0.491	-0.637 ± 0.138	-3.622 ± 3.38	-0.552 ± 0.299	0.086 ± 0.134
DeepSurv	-0.495 ± 0.01	-0.435 ± 0.025	0.414 ± 0.04	-0.249 ± 0.018	0.573 ± 0.042
SuMo-net	-0.376 ± 0.008	-0.391 ± 0.027	1.553 ± 0.132	-0.142 ± 0.042	0.658 ± 0.027

Table 10: Log-likelihood scores for $d = 8$

A.6 Other experiments

Dataset Method	$\mathcal{S}_{C^{td}} \uparrow$				
	FLCHAIN	GBSG	KKBOX	METABRIC	SUPPORT
CoxCC	0.792 ± 0.005	0.672 ± 0.014	0.832 ± 0.001	0.642 ± 0.03	0.608 ± 0.011
Cox-Linear	0.792 ± 0.006	0.66 ± 0.014	0.795 ± 0.001	0.633 ± 0.022	0.596 ± 0.012
Cox-Time	0.792 ± 0.005	0.678 ± 0.017	0.848 ± 0.008	0.649 ± 0.026	0.613 ± 0.007
DeepHit	0.796 ± 0.004	0.683 ± 0.013	0.799 ± 0.005	0.692 ± 0.022	0.642 ± 0.003
DeepSurv	0.791 ± 0.004	0.671 ± 0.014	0.828 ± 0.001	0.645 ± 0.02	0.609 ± 0.011
SuMo-net	0.79 ± 0.004	0.671 ± 0.013	0.76 ± 0.045	0.658 ± 0.029	0.603 ± 0.004
SODEN	N/A	N/A	N/A	0.661 ± 0.004	0.624 ± 0.004
SurvNODE	N/A	N/A	N/A	0.667	0.622

Table 11: Concordance scores. Higher is better

Dataset Method	$\mathcal{S}_{IBS} \downarrow$				
	FLCHAIN	GBSG	KKBOX	METABRIC	SUPPORT
CoxCC	0.112 ± 0.014	0.185 ± 0.006	0.118 ± 0.001	0.179 ± 0.013	0.194 ± 0.003
Cox-Linear	0.102 ± 0.004	0.188 ± 0.004	0.138 ± 0.0	0.176 ± 0.005	0.197 ± 0.002
Cox-Time	0.114 ± 0.017	0.183 ± 0.008	0.104 ± 0.001	0.178 ± 0.005	0.196 ± 0.004
DeepHit	0.162 ± 0.079	0.227 ± 0.064	0.16 ± 0.012	0.21 ± 0.037	0.227 ± 0.019
DeepSurv	0.103 ± 0.004	0.183 ± 0.005	0.116 ± 0.001	0.176 ± 0.005	0.194 ± 0.004
SuMo-net	0.101 ± 0.004	0.179 ± 0.005	0.13 ± 0.023	0.163 ± 0.006	0.196 ± 0.003
SODEN	N/A	N/A	N/A	0.162 ± 0.003	0.190 ± 0.001
SurvNODE	N/A	N/A	N/A	0.157	0.198

Table 12: IBS scores. Lower is better

A.7 Implementation Details

We present the implementation details to SuMo-net, including network architecture parameters, optimizer parameters and the code base.

	Inference Times						Training times (1 epoch)					
	CoxCC	Cox linear	Cox-Time	Deephit	DeepSurv	SuMo-net	CoxCC	Cox linear	Cox-Time	Deephit	DeepSurv	SuMo-net
SUPPORT	0.027 ± 0.003	0.026 ± 0.003	3.769 ± 0.056	0.003 ± 0.0	0.023 ± 0.001	0.044 ± 0.0	0.664 ± 0.068	4.565 ± 6.892	7.411 ± 1.273	0.347 ± 0.136	0.525 ± 0.027	3.236 ± 0.026
METABRIC	0.008 ± 0.001	0.008 ± 0.0	1.586 ± 0.003	0.001 ± 0.0	0.008 ± 0.0	0.016 ± 0.0	0.156 ± 0.01	0.153 ± 0.023	2.676 ± 0.02	0.136 ± 0.002	0.147 ± 0.002	0.337 ± 0.01
GBSG	0.008 ± 0.0	0.008 ± 0.0	1.737 ± 0.029	0.002 ± 0.0	0.009 ± 0.001	0.021 ± 0.0	0.194 ± 0.069	0.161 ± 0.028	2.829 ± 0.053	0.147 ± 0.003	0.159 ± 0.013	0.395 ± 0.01
FLCHAIN	0.021 ± 0.0	0.021 ± 0.001	4.794 ± 0.007	0.003 ± 0.0	0.021 ± 0.0	0.041 ± 0.001	0.403 ± 0.027	0.394 ± 0.028	7.997 ± 0.019	0.208 ± 0.002	0.386 ± 0.014	1.605 ± 0.062
KKBOX	1.836 ± 0.021	1.804 ± 0.002	158.141 ± 0.121	0.351 ± 0.048	1.814 ± 0.021	5.806 ± 0.009	990.645 ± 0.817	990.144 ± 1.737	1156.948 ± 0.338	872.381 ± 1.083	987.592 ± 1.365	700.269 ± 0.517

Table 13: Additional experiments detailing run-times: SuMo-net is the fastest in terms of training speed per epoch on larger datasets.

A.7.1 Hyperparameter space

Hyperparameter	Values
Layers	{1, 2,4,5}
Layers (Covariate part)	{1, 2,4,5}
Nodes per layer	{8,16,32}
Nodes per layer (Covariate part)	{8,16,32}
Dropout	[0.0,0.1,0.2,0.3,0.4,0.5]
Weigh decay	{0.4, 0.2, 0.1, 0.05, 0.02, 0.01, 0}
Batch size	{5,10,25,50,100,250}

Table 14: Hyperparameter search space for experiments on Rot. & GBSG, SUPPORT, METABRIC, and FLCHAIN.

Hyperparameter	Values
Layers	{1, 2,4}
Layers (Covariate part)	{4, 6, 8}
Nodes per layer	{8, 16, 32,64}
Nodes per layer (Covariate part)	{128, 256, 512}
Dropout	[0, 0.7]
Weigh decay	{0.4, 0.2, 0.1, 0.05, 0.02, 0.01, 0}
Batch size	{1000,2500,5000}

Table 15: KKBox hyperparameter configurations. (*) denotes parameters found with a two layer network with 128 nodes.

For exact details, we refer to the code base attached.

A.7.2 Toy datasets

We provide here the distributions of the toy datasets. In each of the distributions $X \sim \text{Unif}[0,1]$.

Dataset	$T X$	$C X$
Weibull	Weib(shape = 2 + 6 X)	Exp(mean = 1.5)
Normal	N(100, 6 X)	N(100, 6)
Checkerboard	CB(4, 6 X)	Exp(mean = 1.5)

Table 16: Distributions of the toy datasets Weibull, Normal and Checkerboard. Weib(shape = s) refers to the Weibull distribution with shape s and scale 1. For the definition of CB(4, 6) see below. Parameters are chosen such that appropriate numbers of individuals is censored (see data description main text) and the survival curves of the Normal and Weibull distributions cross (see plots main text).

The distribution CB(4, 6 $|X$) is defined as follows. Let X be on the horizontal axis and let T be on the vertical axis. We split the square $[0, 1]^2$ up in a grid of 6×4 equally large rectangles, where there are 4 columns and 6

rows. If X is in the 1st or 3rd column, then T is distributed uniformly over the 1st, 3rd, or 5th row. If X is in the 2nd or 4th column, then T is distributed uniformly over the 2nd, 4th or 6th row.