Maximum Likelihood Estimation of Flexible Survival Densities with Importance Sampling

Mert Ketenci MK4139@COLUMBIA.EDU
Department of Computer Science
Columbia University
New York, NY, USA

Shreyas Bhave SAB2323@CUMC.COLUMBIA.EDU
Department of Biomedical Informatics
Columbia University
New York, NY,

Noemie Elhadad NOEMIE.ELHADAD@COLUMBIA.EDU
Department of Biomedical Informatics
Columbia University
New York, NY, USA

Adler Perotte AJP2120@CUMC.COLUMBIA.EDU
Department of Biomedical Informatics
Columbia University
New York, NY,

Abstract

Survival analysis is a widely-used technique for analyzing time-to-event data in the presence of censoring. In recent years, numerous survival analysis methods have emerged which scale to large datasets and relax traditional assumptions such as proportional hazards. These models, while being performant, are very sensitive to model hyperparameters including: (1) number of bins and bin size for discrete models and (2) number of cluster assignments for mixture-based models. Each of these choices requires extensive tuning by practitioners to achieve optimal performance. In addition, we demonstrate in empirical studies that: (1) optimal bin size may drastically differ based on the metric of interest (e.g., concordance vs brier score), and (2) mixture models may suffer from mode collapse and numerical instability. We propose a survival analysis approach which eliminates the need to tune hyperparameters such as mixture assignments and bin sizes, reducing the burden on practitioners. We show that the proposed approach matches or outperforms baselines on several real-world datasets.

1. Introduction

Survival analysis is concerned with modeling time-to-event data by estimating the probability that an event will occur at a future time. Time-to-event data differ from other data by censoring; for certain data points, the true event times may be unobserved. Data is often right censored, indicating that the event occurred after the censoring time but the exact time is unknown. In healthcare settings, survival analysis is useful in a wide variety of applications where censoring naturally occurs, including predicting the risk of disease and

death (Viganò et al., 2000; Perotte et al., 2015; Nagpal et al., 2021c) and analyzing clinical trial data (Fleming and Lin, 2000; Faucett et al., 2002).

Traditionally, survival analysis data has been analyzed using models such as Cox proportional hazards or by fitting the time-to-event distribution using a simple, unimodal parametric distribution, such as the Weibull distribution (Cox, 1972).

In recent years, there have been numerous flexible survival analysis methods introduced which relax these assumptions and are performant on a number of real-world datasets (Ishwaran et al., 2008; Katzman et al., 2018; Kvamme et al., 2019; Nagpal et al., 2021b,c). However, this flexibility has often come with the cost of introducing many additional hyperparameters. Thus, these models require extensive exploration of the hyperparameter space to achieve optimal performance, greatly elevating the burden on practitioners.

Existing survival methods that work with individual-level time-to-event predictions can be grouped into three main categories (Haider et al., 2020): (1) parametric (2) semi-parametric, and (3) non-parametric.

Parametric survival methods assume a known probability distribution over time-to-event data, conditioned on certain covariates, and optimize the log-likelihood or the evidence-lower-bound (ELBO). Known probability distributions (e.g. Weibull, Log-Normal) have the shortcoming that they are constrained by particular hazard function shapes and are unimodal. Some methods have aimed to relax distributional assumptions by using discrete, categorical distributions (Lee et al., 2018; Miscouridou et al., 2018), while others have taken the approach of using continuous, flexible distributions using mixture models (Nagpal et al., 2021b; Han et al., 2022). With the added flexibility, these models have been shown to outperform numerous baseline methods with more restrictive assumptions. However, they also introduce additional hyperparameters. For discrete models, there is the added challenge of specifying the appropriate number of bins and bin sizes. For mixture-based models, the number of mixture distributions must be specified and even with a sufficiently large number of mixtures, the models may collapse to local optima during training resulting in pathologies such as mode collapse (Shireman et al., 2016; Makansi et al., 2019). In addition, these models can be numerically unstable (Makansi et al., 2019).

Most commonly used semi-parametric methods, such as Cox proportional hazards (CoxPH) and DeepSurv, are constrained by the proportional hazards assumptions (Cox, 1972; Katzman et al., 2018). Proportional hazards assume that different instances follow the same hazard trajectory, up to a multiplicative constant, which can be too strict for real-world time-to-event data. Recently, Kvamme et al. (2019) proposed extending CoxPH using a flexible non-proportional hazard function. However, their model does not allow for data sub-sampling and requires gradient approximations that are biased.

Random Survival Forests (RSF) is a well-known non-parametric method for survival analysis (Ishwaran et al., 2008). When tuned carefully, RSF performs on par with or better than the most recent state-of-the-art approaches. However, as also indicated by Nagpal et al. (2021c), RSF is sensitive to certain hyperparameters and require careful tuning.

In this paper, we introduce a flexible parametric survival analysis approach that directly models the hazard function to address the above gaps. Our contributions are as follows:

1. We derive a continuous-time non-proportional survival model whose hazard function can take any shape, both over time and covariates.

2. We introduce an importance sampling (IS) method for estimating the gradient of the otherwise intractable full log-likelihood. Our algorithm scales well to large datasets without requiring biased approximations such as sub-sampling risk sets, numerical integration (Butler, 1985; Kvamme et al., 2019; Danks and Yau, 2022) and computationally expensive approaches that require ODE Solvers (Tang et al., 2022).

3. To the best of our knowledge we are the first to propose an unbiased full log-likelihood optimization method for a non-proportional flexible hazard function without using a mixture model.

4. Other than the network architecture, we only have a single hyperparameter which is the number of importance samples, but this hyperparameter is guaranteed to only improve estimates with larger sizes as we show empirically.

5. We carry out in-depth experimental analysis on several real-world datasets, empirically show the advantages of our approach, and demonstrate that it consistently performs well.

Generalizable Insights about Machine Learning in the Context of Healthcare

Survival analysis has numerous applications in healthcare including risk assessment for chronic diseases and analysis of clinical trial data. In recent years, many methods have emerged which scale to large datasets and relax the restrictive assumptions of widely-used approaches such as Cox proportional hazards. However, this added flexibility has come at the cost of introducing many hyperparameters (e.g., bin size, number of bins, number of mixture components) as well as optimization challenges. This increases the burden placed on practitioners to explore the large space of hyperparameters to ensure optimal performance. We sought to eliminate the necessity for these hyperparameters, while also maintaining all the favorable properties of these models (i.e., flexible, scalable, unbiased estimates, optimized via stochastic gradient descent, continuous). Our model is run with default parameters on all datasets and is able to match or outperform existing state of the art methods. We believe this model will ease the burden on practitioners for fitting new datasets. We have made the code publicly available on GitHub with instructions on how to fit our model on any dataset quickly and efficiently.

2. Sensitivity of Existing Methods to Hyperparameters

One class of continuous time models that can approximate any distribution is mixture density networks. A large number of density components, in theory, can represent any distribution (Bishop and Nasrabadi, 2006). Examples in survival analysis include Nagpal et al. (2021b,c); Han et al. (2022). However, mixture models are prone to arrive at locally optimal solutions resulting in mode collapse and poor density estimation (Shireman et al., 2016; Makansi et al., 2019). We further demonstrate this empirically by running simulation studies, the details of which are specified in Figure 1.

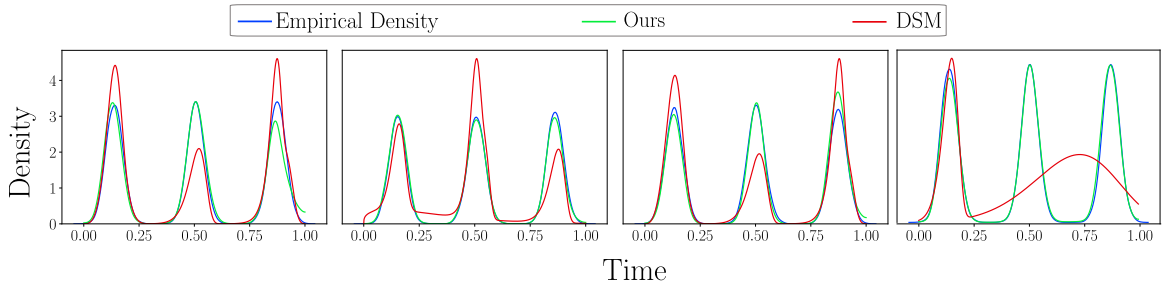

Figure 1: Illustration of our model's ability to capture multi-modal survival densities without requiring a mixture density like Deep Survival Machines (DSM), Deep Cox Mixtures (DCM), and SurvMDN (Nagpal et al., 2021b,c; Han et al., 2022). We simuate a conditional time-to-event dataset using the generative story $t \sim \sum_{j=1}^{3} \frac{1}{3}\mathcal{N}(\mu_j + \boldsymbol{x}_i, 1)$ and $\boldsymbol{x}_i \sim \mathcal{N}(0, 0.01)$ with $\mu_j = 10j$ (we scale the data in $(0, 1]$ to ensure positivity) and fit our model and DSM (with 5 components) over 4 different runs using full log-likelihood optimization. We plot the time-to-event density of a random instance. We observe that the mixture density network can settle in a local-optimum solutions easier. On the other hand, we have a more stable estimation approach which captures the density well across 4 random runs.

Another important class of survival models discretize time with bins for flexible density estimation (Ranganath et al., 2016; Miscouridou et al., 2018; Lee et al., 2018). However, this approach introduces an important hyperparameter: the number of bins. To study the sensitivity of these models to this hyperparameter, we design a simple experiment using DeepHit (Lee et al., 2018) as the model and the commonly used SUPPORT dataset (Knaus et al., 1995) for benchmarking survival models.

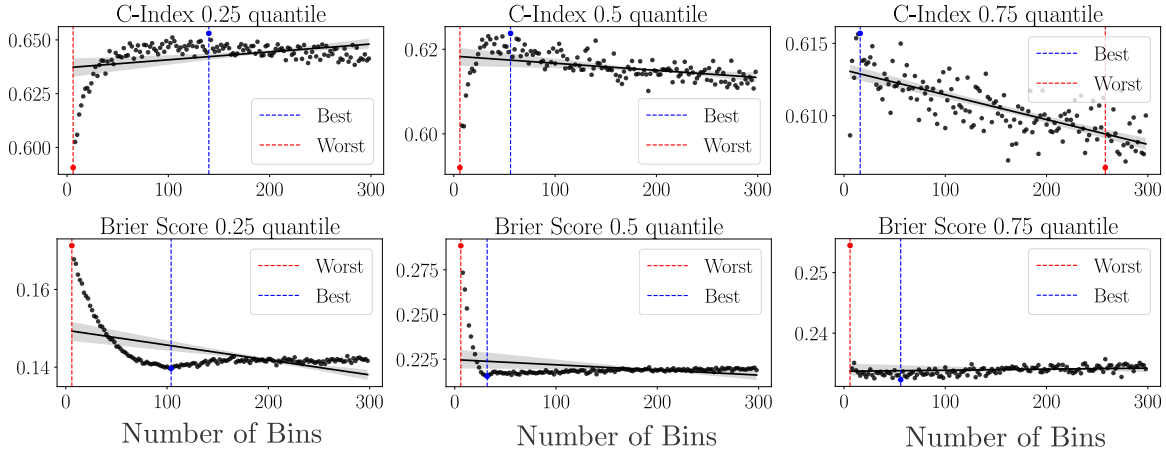

Figure 2: The sensitivity of discrete time models to the number of bins. The blue and red lines describe the best and worst bin sizes for each metric. We employ DeepHit for training and run 750 different runs using 5 fold cross-validation and 150 bin sizes (ranging from 2 to 300 with equal spacing).

Figure 2 shows results averaged across held-out folds for both concordance and brier score as a function of the number of bins. For both metrics, there is an optimal bin number with the metrics performing worse when specifying both a small and very large number of

bins. The optimal number of bins can vary significantly across metrics, making it difficult to specify a single number. This typically results in a trade-off between model calibration and ranking.

3. Deep Hazard Analysis

In this section, we describe our approach called Deep Hazard Analysis (DHA). First, we describe our model in detail in Section 3.1. Then we describe parameter estimation in Section 3.2 and how to make predictions in Section 3.3. Finally, we describe our network architecture in Section 3.4.

3.1. Model

Let $\{\boldsymbol{x}_i, t_i, \delta_i\}_{i=1}^N$ be a right-censored time-to-event data where $\boldsymbol{x}_i \in \mathbb{R}^d$, $t_i \in \mathbb{R}^+$ and $\delta_i \in \{0, 1\}$ stand for covariates, time-to-event, and censoring indicator, respectively. Then we write our probability density function, using the hazard and survival functions, as:

$$f(\boldsymbol{x}, t; \theta) = \underbrace{\lambda(\boldsymbol{x}, t; \theta)}_{\text{Hazard}} \underbrace{S(\boldsymbol{x}, t; \theta)}_{\text{Survival}} \tag{1}$$

$$= \lambda(\boldsymbol{x}, t; \theta) \exp\{-\Lambda(\boldsymbol{x}, t; \theta)\} \tag{2}$$

$$= \lambda(\boldsymbol{x}, t; \theta) \exp\left\{-\int_0^t \lambda(\boldsymbol{x}, t; \theta) dt.\right\}. \tag{3}$$

Here, $\lambda(\boldsymbol{x}, t, \theta) = \log\left(1 + \exp\{\Phi(\boldsymbol{x}, t; \theta)\}\right)$, where $\Phi(.; \theta)$ is a flexible function approximator. In this paper, we use neural networks to model $\Phi(.; \theta)$. The log-likelihood of this survival model for N data points is:

$$\ell = \sum_{i=1}^N \delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) + \log S(\boldsymbol{x}_i, t_i; \theta) dt \tag{4}$$

$$= \sum_{i=1}^N \left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \int_0^{t_i} \lambda(\boldsymbol{x}_i, t; \theta) dt\right), \tag{5}$$

Note that, the hazard function implied by our model is not restricted by the proportional hazards assumption.

3.2. Parameter Estimation

The integral in Equation 5 is intractable. A straight-forward approach to approximate it in an unbiased way is by importance sampling:

$$\ell = \sum_{i=1}^{N} \left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \int_0^{t_i} \lambda(\boldsymbol{x}_i, t; \theta) dt \right) \tag{6}$$

$$= \sum_{i=1}^{N} \left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - t_i \int_0^{t_i} \frac{\lambda(\boldsymbol{x}_i, t; \theta)}{t_i} dt \right) \tag{7}$$

$$= \sum_{i=1}^{N} \left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - t_i \, \mathbb{E}_{t \sim U(0,t_i)} \left[\lambda(\boldsymbol{x}_i, t; \theta) \right] \right) \tag{8}$$

$$= N \, \mathbb{E}_{\mathcal{D}} \left[\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - t_i \, \mathbb{E}_{U(0,t_i)} \left[\lambda(\boldsymbol{x}_i, t; \theta) \right] \right]. \tag{9}$$

Here $U(.)$ is the uniform distribution. Note that the integral in Equation 7 is the expected value of the hazard w.r.t. time, on a uniform grid, allowing for Equation 8. We approximate Equation 9 by drawing L Monte-Carlo samples from the empirical data distribution $\mathcal{D}$ and a set $\tilde{\boldsymbol{T}}_i = \{\tilde{t}_{ij}\}_{j=1}^{M}$ from the IS distribution $U(0, t_i)$. The loss function after using a mini-batch of L data and M IS samples is denoted as:

$$\tilde{\ell}(\boldsymbol{T} | \boldsymbol{X}, \Delta, \tilde{\boldsymbol{T}}; \theta)$$
$$= \frac{N}{L} \sum_{i=1}^{L} \left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \frac{t_i}{M} \sum_{j=1}^{M} \lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta) \right), \tag{10}$$

where $\boldsymbol{T} = \{t_i\}_{i=1}^{L}$, $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{L}$, $\Delta = \{\delta_i\}_{i=1}^{L}$, and $\tilde{\boldsymbol{T}} = \{\tilde{\boldsymbol{T}}_i\}_{i=1}^{L}$. We describe our learning algorithm in Algorithm 1.

Algorithm 1 Mini-batch stochastic gradient descent algorithm for our model.

Input: $\mathcal{D}$
$\theta \leftarrow$ Initialize parameters
while not converged **do**
  $\{\boldsymbol{X}, \boldsymbol{T}, \Delta\} \leftarrow$ Sample L data from $\mathcal{D}$
  $\tilde{\boldsymbol{T}} \leftarrow$ Sample M importance samples from $U(0, \boldsymbol{T})$
  $g \leftarrow \nabla_\theta \tilde{\ell}(\boldsymbol{T} | \boldsymbol{X}, \Delta, \tilde{\boldsymbol{T}}; \theta)$
  $\theta \leftarrow$ Update using gradients g
end while
Output: θ

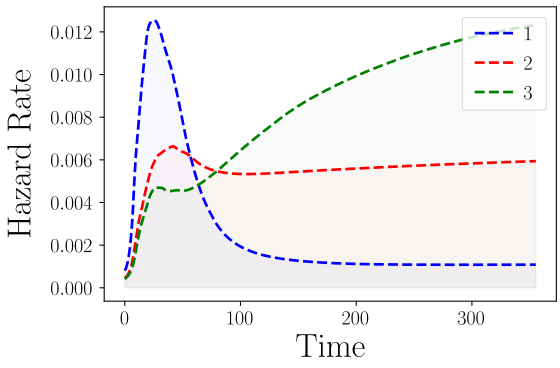

Figure 3: Illustration of our model's ability to capture non-proportional hazard functions on 3 random instances from the METABRIC dataset. Blue, green, and red curves denote different patients. Unlike proportional models, such as CoxPH and DeepSurv, the hazard rate and its shape differ between patients. Patient 1 is under high instantaneous risk in the early stages, which later decreases significantly. The instantaneous risk for patient 3 increases by time while patient 2 remains relatively constant. Our model does not need for critical hyperparameters such as cluster size, discretization or ODESolvers to model non-linear hazard rates.

Importance sampling can be problematic in high-dimensional spaces with an exponential growth in variance (Scharth and Kohn, 2016). However, we are strictly interested in modeling 1-dimensional time-to-event data. We demonstrate the stability and convergence of our learning process in Figure 5. Moreover, the support of the uniform distribution is well-defined over 0 and t_i eliminating the need for rejecting any sample.

Unbiased gradients. An important property that distinguishes our approach from Kvamme et al. (2019) is that Equation 10 allows for unbiased learning of θ. Hence, $\nabla_\theta \tilde{\ell}(\boldsymbol{T}|\boldsymbol{X}, \Delta, \tilde{\boldsymbol{T}}; \theta)$ is an unbiased Monte-Carlo estimate of the true gradients, $\nabla_\theta \ell$:

$$\mathbb{E}_{\mathcal{D}}\left[\nabla_\theta \tilde{\ell}(\boldsymbol{T}|\boldsymbol{X}, \Delta, \tilde{\boldsymbol{T}}; \theta)\right] = \mathbb{E}_{\mathcal{D}}\left[\nabla_\theta \frac{N}{L}\sum_{i=1}^{L}\left(\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \frac{t_i}{M}\sum_{j=1}^{M}\lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)\right)\right] \tag{16}$$

$$= \nabla_\theta \frac{N}{L}\sum_{i=1}^{L}\left(\mathbb{E}_{\mathcal{D}}\left[\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \frac{t_i}{M}\sum_{j=1}^{M}\lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)\right]\right) \tag{17}$$

$$= \nabla_\theta \frac{N}{L} L\, \mathbb{E}_{\mathcal{D}}\left[\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - \frac{1}{M}\sum_{j=1}^{M} t_i \lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)\right] \tag{18}$$

$$= \nabla_\theta \underbrace{N\, \mathbb{E}_{\mathcal{D}}\left[\delta_i \log \lambda(\boldsymbol{x}_i, t_i; \theta) - t_i\, \mathbb{E}_{U(0,t_i)}\left[\lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)\right]\right]}_{\ell} \tag{19}$$

$$= \nabla_\theta \ell. \tag{20}$$

which requires evaluating a set $\tilde{\boldsymbol{T}}_i$ of M importance samples with linear $\mathcal{O}(M)$ time complexity as shown in Algorithm 2.

The time complexity for a set of N instances can be thought of $\mathcal{O}(NM)$. However, GPUs allow for parallel computing over instances which practically results in $\mathcal{O}(M)$ operations for small IS sizes which empirically show this in Figure 4. $\tilde{S}(t_i, \boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta) \xrightarrow{p} S(t_i, \boldsymbol{x}_i; \theta)$ as M increases, therefore it is beneficial to work with a relatively large M. We study the implications of IS to predictions empirically in Section 5.3.

3.3. Predictions

The quantity of interest is the probability of survival of an instance above some point in future t_i denoted by $S(t_i|\boldsymbol{x}_i;\theta)$, which is analytically intractable. The IS method can be employed here to predict the survival as:

$$S(t_i, \boldsymbol{x}_i; \theta) \tag{11}$$

$$= \exp\left\{-\int_0^{t_i} \lambda(\boldsymbol{x}_i, t; \theta)dt\right\} \tag{12}$$

$$= \exp\left\{-t_i\, \mathbb{E}_{U(0,t_i)}\left[\lambda(\boldsymbol{x}_i, t; \theta)\right]\right\} \tag{13}$$

$$\approx \exp\left\{-\frac{t_i}{M}\sum_{j=1}^{M} \lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)\right\} \tag{14}$$

$$= \tilde{S}(t_i, \boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta), \tag{15}$$

This is important because it guarantees that the model parameters θ converge to a value that optimizes the full log-likelihood ℓ while scaling to large datasets by data subsampling.

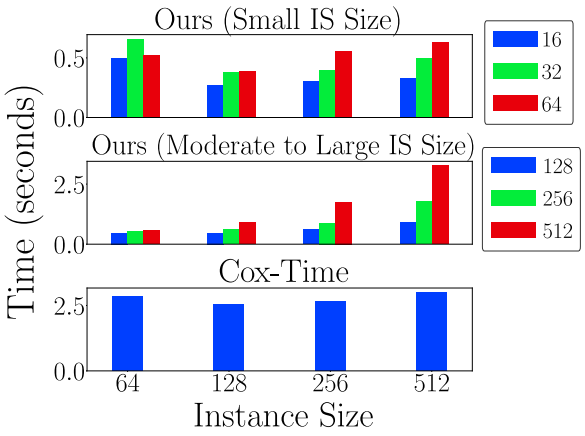

Figure 4: The time required to predict different IS and patient sizes on the METABRIC dataset. Blue, green, and red bars denote the IS sizes. Empirically, we observe that prediction time scales linearly with respect to IS and instance sizes for moderate to large IS sizes. This result is as expected. For IS sizes of 16, 32, 48, and 64 altering the instance size does not result in a major difference in computational time as discussed in Section 3.3. We also show the Cox-Time model to provide a baseline for the prediction times. We did not add the time required to fit the baseline hazard function for the Cox-Time model. Our implementation is approximately 5 times faster than the Cox-Time model for IS size ≤ 64. For a fair comparison, we use the same neuron, layer sizes, and GPU for both models. The numbers are reported for single-precision floating-point format.

Algorithm 2 Making predictions.

Input: $\boldsymbol{x}_i, t_i, \theta$
$\tilde{\boldsymbol{T}}_i \leftarrow$ Sample M importance samples from $U(0, t_i)$
$\tilde{\Lambda}(\boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta) \leftarrow$ Calculate $\frac{t_i}{M}\sum_{j=1}^{M} \lambda(\boldsymbol{x}_i, \tilde{t}_{ij}; \theta)$
$\tilde{S}(t_i, \boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta) \leftarrow \exp\left\{-\tilde{\Lambda}(\boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta)\right\}$
Output: $\tilde{S}(t_i, \boldsymbol{x}_i, \tilde{\boldsymbol{T}}_i; \theta)$

3.4. Network Architecture

We experiment with two neural network architectures to parameterize our model. Architecture 1 (A1) is formulated by:

$$\Phi(\boldsymbol{x}, t; \theta) = \Phi_{\text{shared}}(\text{cat}[\boldsymbol{x}, t]; \theta_{\text{shared}}). \tag{21}$$

and (A2) is formulated by:

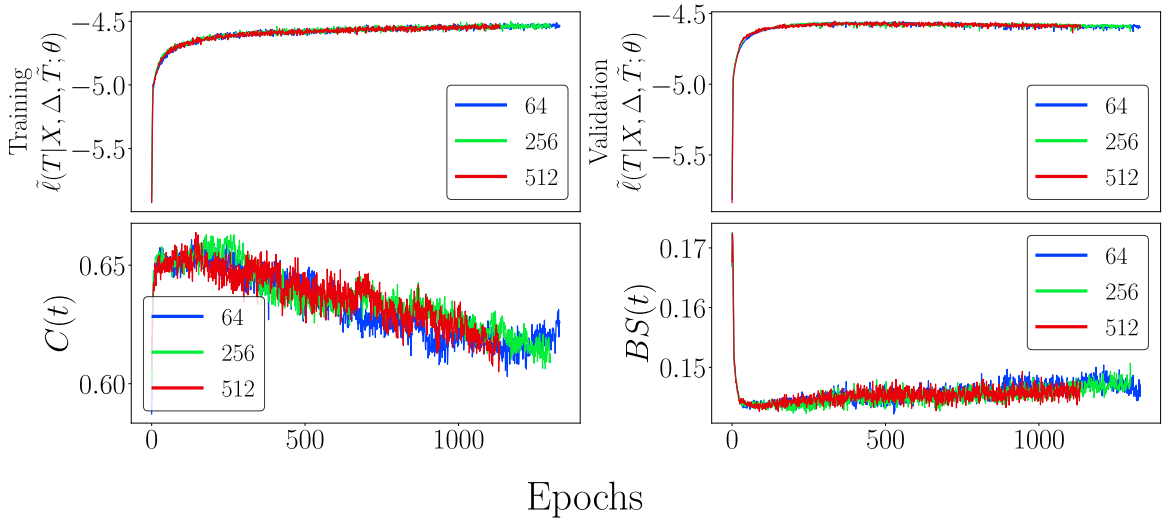

Figure 5: Training and validation log-likelihood, C-Index ($C(t)$) and Brier Score ($BS(t)$), for $t : p(T < t) \leq 1/4$ (25^{th} quantile), by epochs of our approach for different IS sizes. Blue, green, and red lines denote the IS sizes. Training and validation log-likelihoods increase monotonically. Training stops when the validation log-likelihood does not improve for 800 epochs. Interestingly, lower IS sizes do not add a substantial variance and change the values the algorithm converges, for both training and validation log-likelihood values. The survival metrics for different IS sizes are also similar across different datasets, which empirically shows our algorithm's stability. We show this empirically for two different network architectures over three different importance sampling sizes on 4 real-world datasets, empirically. The empirical results are discussed in Section 6.

$$\begin{aligned}&\Phi(\boldsymbol{x}, t; \theta)\\ &= \Phi_{\text{shared}}(\text{cat}[\Phi_{\text{cov}}((\boldsymbol{x}; \theta_{\text{cov}}), \Phi_{\text{time}}(t; \theta_{\text{time}})]; \theta_{\text{shared}}).\end{aligned} \tag{22}$$

Here cat operation refers to concatenation of two vectors. Later, the output of the neural networks are feed into the softplus function to predict the hazard rate. The intuition behind the second architecture is to allow for learning a temporal embedding independent of patient covariates. The architectures are shown in Figure 6.

4. Related Work

Parametric methods. Deep Survival Analysis (DSA) and Deep Survival Machines (DSM) are two important examples of parametric methods (Ranganath et al., 2015; Nagpal et al., 2021b). DSA models time-to-event data using Weibull distribution conditioned on a latent representation drawn from a deep exponential family. A limitation of this approach is that it assumes proportional hazards with a Weibull base hazard rate for a fixed shape parameter. DSM models time-to-event data using a mixture of Log-normal and Weibull densities whose parameters are conditioned on individual instances and optimize the ELBO. Although a mixture of such densities allows for a more flexible hazard, the mixture size introduces an additional hyperparameter to tune.

Another important parametric line of work focuses on discretizing time. DeepHit is a well-known approach in this line of work (Lee et al., 2018, 2019a). This approach divides

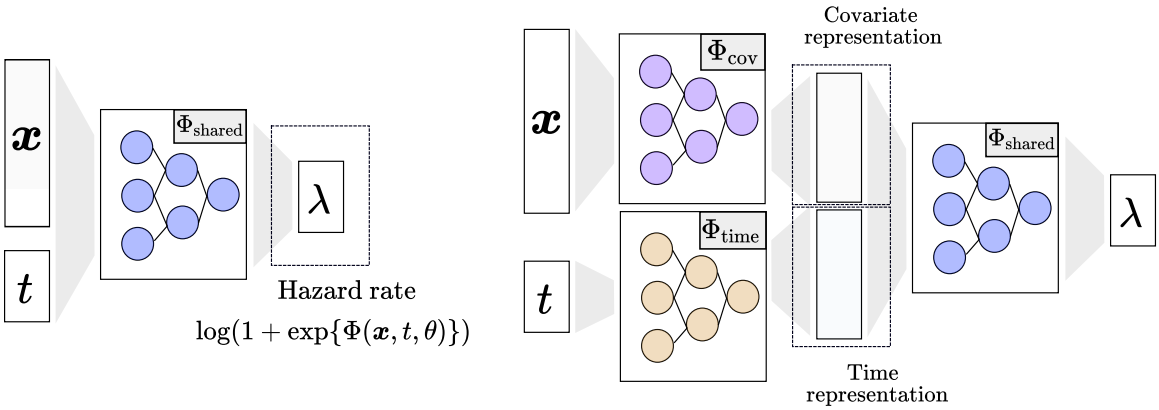

Figure 6: The network architectures A1 (left) and A2 (right). A1 is a simple feed forward network parameterized by θ_{shared}. A2 has two separate input layers for covariates $\boldsymbol{x}$ and time t. These layers form a latent representation of $\boldsymbol{x}$ and t that are concatenated and fed into a shared layer followed by softplus function to model the hazard rate, λ.

continuous time-to-event data into bins and assumes a categorical distribution. A limitation of this line of work is that it is sensitive to the heuristically chosen bin size. Additionally, it only allows for making predictions over a pre-defined set of time bins which can be problematic for instances that do not fall into this range requiring additional interpolations.

Semi-parametric methods. The Cox proportional hazards model (CoxPH) is commonly employed in various settings due to its simplicity (Cox, 1972). A significant amount of work focused on improving CoxPH. Rosen and Tanner (1999) improved CoxPH using a mixture of linear experts. Nagpal et al. (2019) extended this approach using a variational objective.

Other approaches that extend CoxPH involve using flexible function approximators. The Faraggi-Simon network was the first to couple CoxPH with neural networks (Faraggi and Simon, 1995). However, this attempt did not demonstrate improvements over CoxPH (Mariani et al., 1997; Xiang et al., 2000; Sargent, 2001). Later, Katzman et al. (2018) showed that modern deep-learning techniques improve CoxPH. A shortcoming of these approaches is the proportional hazards assumption, which can be too restrictive for real-world datasets.

More recently, Deep Cox Mixtures (DCM) proposed to use a mixture of non-linear Cox experts with the EM algorithm and further improved upon DeepSurv Nagpal et al. (2021c). There have also been methods that extended Cox's framework to handle unstructured data, such as images (Zhu et al., 2016, 2017; Christ et al., 2017).

Non-parametric methods. RSF is an important method in this line of work (Ishwaran et al., 2008). RSF aggregates multiple trees by bagging and averages the result of each tree when making predictions. A limitation of the RSF is that the current implementations do not support GPUs, and it is unclear how to deploy them for datasets with large examples and covariates.

Amongst recent work, Cox-Time is the closest approach to DHA (Kvamme et al., 2019). Cox-Time extends CoxPH by removing the proportional hazards assumption and leveraging neural networks. Cox-Time is a semi-parametric model that optimizes partial log-likelihood.

In its original form, Cox-Time is amenable to stochastic optimization and does not scale well to large datasets. Therefore, learning Cox-Time requires a biased but computationally cheaper gradient approximation. Unlike Cox-Time, our model (1) is parametric, (2) optimizes the full log-likelihood, and (3) does not require biased gradient approximations.

5. Experiments

In this section, we describe the datasets (5.1), baseline models (5.2) and evaluation metrics (5.3) used to asses the performance of our model. We compare our model to multiple state-of-the-art approaches on three commonly used real-world datasets. We also investigate the impact of different IS sizes on these datasets.

5.1. Datasets

SUPPORT. Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) consists of seriously ill hospitalized adult patients (Knaus et al., 1995). After preprocessing, there are 8873 instances and 23 covariates with median follow-up days and censoring rate of 231 and 31.9%, respectively. We use the preprocessing of the PyCox library (Kvamme, 2022).

METABRIC. A Canada-UK project, the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database, comprises targeted sequencing and survival information from breast cancer patients (Curtis et al., 2012; Pereira et al., 2016). After preprocessing, the dataset contains 1904 instances and 9 covariates with median follow-up days and a censoring rate of 114.9 days and 42%, respectively. We use the preprocessing of the PyCox library (Kvamme, 2022).

FLCHAIN. A controlled trial conducted in Olmsted County, Minnesota that investigates the association of mortality and assay of serum free light chain (FLCHAIN) (Dispenzieri et al., 2012). After preprocessing, there are 6524 instances with 16 covariates. The median follow-up days and a censoring rate of 4303 days and 70%, respectively. We use the preprocessing of the PyCox library (Kvamme, 2022).

CKD/AKI. The study cohort consisted of 10,173 patients that are identified in chronic kidney disease (CKD) incident cohort where the event is defined as acute kidney injury (AKI) diagnosis, during hospitalization, by the Health Equity Research Assessment (HERA) characterization. The median follow-up days and censoring rate are 67 days and 64%, respectively.

5.2. Baseline Models

Deep Survival Machines (DSM). A parametric survival model that extends beyond the AFT using a mixture of Weibull and log-normal distributions. The mixture assignments and time-to-event distributions are parameterized by neural networks conditioned on covariates. Parameter estimation is done by optimizing the ELBO, where the expectation is taken with respect to the conditional model prior (Nagpal et al., 2021b).

Deep Cox Mixtures (DCM). A semi-parametric survival model that extends DeepSurv using a mixture of CoxPH components parameterized by neural networks. Parameter estimation is done by the EM algorithm and fitting polynomial splines to baseline hazards (Nagpal et al., 2021c).

DeepSurv. A semi-parametric model that extends CoxPH by modeling relative risk using neural networks conditioned on covariates formulating a non-linear proportional hazard function(Katzman et al., 2018).

DeepHit. A discrete-time survival model parameterized by neural networks with a softmax output layer. DeepHit uses cross-entropy loss combined with a ranking loss (Lee et al., 2018).

Random Survival Forest (RSF). An extension of random forests that fits multiple trees to survival data by bagging and using the cumulative hazard function computed by the Nelson-Aalen estimator (Ishwaran et al., 2008).

Cox-Time. A semi-parametric method that extends CoxPH beyond propotional hazards. Cox-Time uses neural networks to parameterize the hazard function (Kvamme et al., 2019). Parameter estimation is done by optimizing a biased approximation of the partial log-likelihood.

CoxPH. Well-known semi-parametric Cox proportional hazards model (Cox, 1972). Parameter learning is done by optimizing the partial log-likelihood.

5.3. Evaluation Metrics

The literature on evaluation metrics for survival analysis is vast and beyond this paper's scope. We refer the reader to references in the following paragraphs for more detail. This paper focuses on evaluations over a fixed follow-up period. Fixed follow-up periods are used in many important real-world settings, such as randomized clinical trials. Similar to Li et al. (2023); Nagpal et al. (2022a, 2021b,c); Jeanselme et al. (2022); Wang and Sun (2022); Lee et al. (2019b), we consider event quantiles as follow-up periods and report concordance index (C-Index), Brier score (BS), and Area Under Receiver Operating Curve (ROC-AUC) metrics which are implemented by Pölsterl (2020). This provides an overview of how each model performs over time and helps ensure that the models effectively capture potential differences in risks over the event horizon. All metrics are adjusted by the inverse probability of censoring weight (IPCW), using the Kaplan-Meier censoring estimate, to account for the censoring bias (Kaplan and Meier, 1958).

Concordance Index (C-Index). C-Index is the probability that predicted survival durations for two instances have the same ordering as their actual survival times. Initially, C-Index was derived for proportional hazards framework by Harrell et al. (1982) and later extended for non-proportional cases (Antolini et al., 2005). More recently, Uno et al. (2011) proposed to correct C-Index for censoring using IPCW which is what we employ in this paper:

$$\mathrm{C}(t) = P(S(t|\boldsymbol{x}_i) < S(t|\boldsymbol{x}_j)|t_i < t_j, t_i < t). \tag{23}$$

SUPPORT Dataset									
Models	25^{th} Quantile			50^{th} Quantile			75^{th} Quantile		
	C-Index	BS	ROC	C-Index	BS	ROC	C-Index	BS	ROC
CoxPH	0.553	0.262	0.558	0.567	0.222	0.587	0.590	0.351	0.649
DeepSurv	0.603	0.143	0.607	0.598	0.217	0.618	0.610	**0.231**	**0.661**
RSF	**0.657**	**0.140**	**0.664**	0.621	0.215	0.638	**0.613**	**0.232**	**0.666**
DeepHit	0.631	0.153	0.638	0.608	0.243	0.628	0.607	0.236	**0.659**
Cox-Time	0.640	**0.140**	0.647	0.620	0.213	0.640	**0.615**	**0.231**	**0.663**
DSM	0.645	**0.140**	0.650	0.621	0.214	0.637	**0.615**	0.245	**0.657**
DCM	**0.649**	**0.140**	**0.656**	0.616	0.216	0.633	0.602	0.236	0.650
DHA (IS = 64, A1)	**0.651**	**0.139**	**0.660**	0.623	0.213	0.640	**0.614**	0.236	0.644
DHA (IS = 256, A1)	**0.652**	**0.140**	**0.660**	0.624	0.213	0.641	**0.614**	0.236	0.646
DHA (IS = 512, A1)	**0.652**	**0.139**	**0.661**	0.624	0.213	0.641	**0.615**	0.235	0.647
DHA (IS = 64, A2)	**0.659**	**0.139**	**0.667**	**0.629**	**0.212**	**0.647**	**0.615**	0.234	0.653
DHA (IS = 256, A2)	**0.659**	**0.139**	**0.667**	**0.629**	**0.212**	**0.648**	**0.615**	0.234	0.654
DHA (IS = 512, A2)	**0.660**	**0.139**	**0.667**	**0.629**	**0.212**	**0.647**	**0.615**	0.234	0.654

Table 1: The results on the SUPPORT dataset. For C-Index and ROC, higher scores are better. For BS, lower is better. The best mean results are **underlined** and the results that are close to the best one, by repeated k-fold cv t-test statistics in 95% confident interval, are shown in **bold**.

Brier Score (BS). A perfect ranking can be obtained without assessing appropriate risk scores, which results in calibration problem. BS measures the model calibration by the expected square difference between the survival predictions and event indicators:

$$\text{BS}(t) = \mathbb{E}\left[(I_{t_i>t} - S(t|\boldsymbol{x}_i))^2\right]. \tag{24}$$

BS is originally derived to evaluate the accuracy of weather forecasts by Brier et al. (1950) and later extended to censored time-to-event datasets by Graf et al. (1999) which is what we employ in this paper.

Area Under Receiver Operating Curve (ROC-AUC). ROC-AUC quantifies the seperation of positive and negative instances where the positives are defined as the instances that experienced the event before time t:

$$\text{AUC}(t) = P(S(t|\mathbf{x}_i) \leq S(t|\mathbf{x}_j)|t_i \leq t, t_j > t) \tag{25}$$

This definition also relates to the time-dependent C-Index derived by Antolini et al. (2005), which is based on the sum of weighted AUC scores at different time steps. Similar to previous metrics, we adjust this measure for censoring to have an unbiased estimate (Hung and Chiang, 2010; Kamarudin et al., 2017).

5.4. Experimental Design

We perform 2x5-fold cross-validation (cv) for our model and baselines. The random seeds are fixed, and each train-valid-test splits seen by the models are identical for all datasets within runs. We use t-test with corrected repeated k-fold cv test to correct our t-statistics for the correlation between splits (Bouckaert and Frank, 2004):

Models	METABRIC Dataset								
	25^{th} Quantile			50^{th} Quantile			75^{th} Quantile		
	C-Index	BS	ROC	C-Index	BS	ROC	C-Index	BS	ROC
CoxPH	0.629	0.244	0.640	0.627	0.196	0.649	0.633	0.334	0.684
DeepSurv	0.640	0.122	0.653	0.636	0.197	0.657	0.635	0.227	0.677
RSF	**0.702**	**0.117**	0.718	**0.669**	**0.192**	**0.689**	**0.639**	0.227	0.675
DeepHit	**0.703**	**0.116**	**0.719**	0.653	0.194	0.674	0.616	0.230	0.665
Cox-Time	**0.703**	**0.117**	**0.720**	0.665	**0.191**	0.688	**0.638**	**0.226**	**0.681**
DSM	**0.701**	**0.118**	0.715	0.667	0.209	0.685	**0.641**	0.261	**0.672**
DCM	**0.698**	0.122	0.714	0.663	0.200	0.682	**0.637**	0.232	**0.673**
DHA (IS = 64, A1)	**0.706**	**0.117**	**0.720**	**0.670**	**0.192**	**0.690**	**0.647**	**0.226**	**0.686**
DHA (IS = 256, A1)	**0.708**	**0.117**	**0.722**	**0.670**	**0.192**	0.688	**0.643**	0.227	**0.683**
DHA (IS = 512, A1)	**0.706**	**0.117**	**0.721**	0.668	0.193	0.688	**0.643**	0.227	**0.684**
DHA (IS = 64, A2)	**0.710**	**0.117**	**0.725**	**0.672**	**0.192**	**0.692**	0.637	0.227	**0.677**
DHA (IS = 256, A2)	**0.712**	**0.116**	**0.726**	**0.675**	**0.191**	**0.695**	**0.639**	**0.225**	**0.680**
DHA (IS = 512, A2)	**0.710**	**0.116**	**0.725**	**0.676**	**0.190**	**0.696**	**0.643**	**0.224**	**0.684**

Table 2: The results on the METABRIC dataset. For C-Index and ROC, higher scores are better. For BS, lower is better. The best mean results are **underlined** and the results that are close to the best one, by repeated k-fold cv t-test statistics in 95% confident interval, are shown in **bold**.

$$t^l = \frac{\mu^l}{\hat{\sigma}^{l^2}\sqrt{\frac{1}{kr} + \frac{n_{\text{te}}}{n_{\text{tr}}}}}. \tag{26}$$

Here, $\mu^l = \frac{1}{kr}\sum_{i=1}^{k}\sum_{j=1}^{r} y_{ij}^l$, where y_{ij}^l corresponds to performance difference between two models for i^{th} fold and j^{th} run on l^{th} metric. k and r are defined as number of folds and runs, and n_{tr} and n_{te} are train and test sizes, respectively. Finally, $\hat{\sigma}^{l^2} = \frac{1}{kr-1}\sum_{i=1}^{k}\sum_{j=1}^{r}(y_{ij}^l - \mu^l)$. The test statistic t^l is distributed according to Student's t-distributions with $kr-1$ degrees of freedom.

We report the mean results for 25^{th}, 50^{th}, and 75^{th} quantiles for each metric, and highlight them with respect to the t-statistic described above.

Each baseline model has been fully tuned for each dataset using the validation set. The models are trained for 4000 epochs until convergence on each fold and run. To ensure a fair comparison, we perform early stopping on all models using their validation loss. We changed the IS size while using fixed hyperparameters in our model to study the effects of IS. We describe the hyperparameter optimization protocol in Appendix 8. We use a single Nvidia GeForce RTX 20 series graphics card to carry-out our experiments. We refer reader to Nagpal et al. (2022b), Kvamme (2022), and Pölsterl (2020) for baseline implementations.

6. Results and Discussion

For the SUPPORT dataset, our approach yields the best performance results over 25^{th} and 50^{th} event quantiles. For the 75^{th} quantile RSF and Cox-Time are on par, while we yield the best results on C-Index.

FLCHAIN Dataset									
Models	25^{th} Quantile			50^{th} Quantile			75^{th} Quantile		
	C-Index	BS	ROC	C-Index	BS	ROC	C-Index	BS	ROC
CoxPH	0.789	0.103	0.800	**0.793**	**0.098**	**0.816**	**0.791**	0.168	**0.826**
DeepSurv	0.786	**0.060**	0.797	0.790	**0.100**	0.813	0.788	**0.126**	**0.823**
RSF	**0.801**	**0.058**	**0.813**	**0.796**	**0.098**	**0.819**	**0.792**	**0.124**	**0.827**
DeepHit	0.792	0.061	0.803	**0.794**	**0.101**	**0.817**	**0.790**	**0.127**	**0.825**
Cox-Time	**0.795**	0.066	**0.807**	**0.796**	0.120	**0.819**	**0.792**	0.165	**0.827**
DSM	**0.791**	0.061	0.803	**0.793**	0.111	**0.815**	**0.790**	0.147	**0.825**
DCM	**0.793**	**0.059**	**0.805**	0.785	**0.101**	0.806	**0.780**	0.128	0.813
DHA (IS = 64, A1)	**0.793**	0.063	0.804	0.792	0.109	0.814	**0.790**	0.143	0.822
DHA (IS = 256, A1)	**0.793**	0.063	0.804	**0.793**	0.110	**0.815**	0.789	0.145	0.822
DHA (IS = 512, A1)	**0.793**	0.063	0.803	**0.793**	0.109	0.814	0.789	0.143	0.822
DHA (IS = 64, A2)	**0.799**	0.061	**0.810**	**0.795**	0.105	**0.817**	**0.790**	0.138	0.823
DHA (IS = 256, A2)	**0.799**	0.062	**0.811**	**0.794**	0.106	**0.816**	0.788	0.140	0.821
DHA (IS = 512, A2)	**0.800**	0.061	**0.812**	**0.795**	0.105	**0.817**	0.789	0.139	0.822

Table 3: The results on the FLCHAIN dataset. For C-Index and ROC, higher scores are better. For BS, lower is better. The best mean results are **underlined** and the results that are close to the best one, by repeated k-fold cv t-test statistics in 95% confident interval, are shown in **bold**.

For the METABRIC dataset, Cox-Time, RSF and our approach perform well across shorter and longer time-horizons with our approach having the best average results over shorter and longer horizons for different importance sampling sizes.

For the FLCHAIN dataset, we see that RSF retains the best average result while our approach is comparable on ranking based metrics.

Finally, for CKD/AKI dataset, our approach retains the best mean results across most of the metrics over both shorter and longer time horizons while DeepHit being our closest competitor.

Overall, our approach consistently demonstrates better results in **29** out of 36 dataset-metric pairs with **21** out of **29** being the best average, over different baseline models, including continuous and discrete state-of-the-art approaches. Our closest competitor is RSF, which demonstrates better results in **30** out of 36 metrics with **12** out of 30 being the best average.

To summarize our results, we emphasize several important points: **(1)** For neural non-proportional hazard modeling having a separate embedding layer (A2) for time is more beneficial than concatenating and feeding everything to a shared neural network (A1), **(2)** despite being introduced much earlier, when tuned carefully, RSF performs on par or better than the other models. **(3)** best-performing benchmark models differ between datasets and time-horizons, confirming the findings of Lee et al. (2019a), **(4)** our model performs well consistently over different datasets and time horizons with minimal hyperparameter tuning. **(5)** parametric continuous-time models are more robust to hyperparameter choice while discrete-time models (e.g., DeepHit) have critical hyperparameters as also emphasized by

CKD/AKI Dataset									
Models	25^{th} Quantile			50^{th} Quantile			75^{th} Quantile		
	C-Index	BS	ROC	C-Index	BS	ROC	C-Index	BS	ROC
CoxPH	0.598	0.089	0.612	0.581	0.167	0.614	0.575	0.225	0.649
DeepSurv	0.619	**0.088**	0.633	0.607	0.164	0.643	0.603	**0.219**	0.673
RSF	**0.639**	**0.087**	**0.655**	**0.620**	**0.163**	**0.657**	**0.608**	**0.218**	**0.686**
DeepHit	**0.642**	**0.087**	**0.658**	**0.624**	**0.162**	**0.663**	**0.610**	**0.218**	**0.686**
Cox-Time	0.623	**0.089**	0.635	0.613	0.165	0.649	**0.605**	0.220	0.676
DSM	**0.635**	**0.088**	0.647	**0.615**	0.182	**0.658**	0.602	0.247	**0.688**
DHA (IS = 64, A1)	**0.637**	**0.087**	0.650	**0.617**	**0.163**	0.653	**0.605**	0.220	**0.685**
DHA (IS = 256, A1)	**0.638**	**0.088**	0.652	**0.618**	**0.163**	0.654	**0.605**	0.220	0.682
DHA (IS = 512, A1)	**0.637**	**0.088**	0.649	**0.617**	**0.163**	0.653	**0.605**	0.221	**0.683**
DHA (IS = 64, A2)	**0.644**	**0.087**	**0.660**	**0.625**	**0.162**	**0.661**	**0.610**	**0.218**	**0.685**
DHA (IS = 256, A2)	**0.646**	**0.087**	**0.660**	**0.625**	**0.162**	**0.659**	**0.611**	**0.218**	**0.683**
DHA (IS = 512, A2)	**0.642**	**0.087**	**0.657**	**0.623**	0.164	**0.659**	**0.608**	**0.218**	**0.683**

Table 4: The results on the CKD/AKI dataset. For C-Index and ROC, higher scores are better. For BS, lower is better. The best mean results are **underlined** and the results that are close to the best one, by repeated k-fold cv t-test statistics in 95% confident interval, are shown in **bold**. We found DCM to be unstable on this dataset and did not include it.

Sloma et al. (2021).[1] **(6)** altering the IS size does not result in a significant change, which shows the robustness of our approach.

7. Limitations and Future Work

We consider a number of limitations for this work to address in the future:

Competing Risk Scenarios. We consider extending our approach for competing-risk scenarios in which various events may lead to failure. In particular, modifying our architecture to accommodate competing risks utilizing a common covariate layer and sub-networks for each competing event, and adjusting the likelihood may enable information flow from various risks an instance confronts. This approach is similar to DeepHit (Lee et al., 2018).

Temporal Data. Certain time-to-event data, such as vital signs and electronic health records, can consist of time series. In such cases, leveraging the temporal structure of data is important. Similar to Nagpal et al. (2021a), we consider altering the network architecture to account for the temporality of the clinical data using recurrent neural networks (RNNs).

Different Modalities. Another potential direction for this work includes using different data modalities to perform survival analysis, such as medical imaging. We consider altering the network to incorporate image data using Convolutional Neural Networks (CNNs) like Zhu et al. (2016).

1. In particular, we found that DeepHit is sensitive to the number of output neurons ('num_durations') and must be tuned carefully: too many results in the training of very few, and too few result in information loss.

Small Datasets. DHA is a deep learning model. We acknowledge that deep learning models require large datasets and may demonstrate inferior performance compared to non-parametric approaches, like RSF, when dealing with small datasets.

8. Conclusion

In this work, we demonstrate that there are a number of undesirable characteristics in existing state of the art survival models. In particular, discrete time models and mixture density models are very sensitive to hyperparameters such as number of bins and mixtures. Each of these choices requires extensive tuning by practitioners to achieve optimal performance. We introduce a method which is free of such hyperparameters and exhibits all the desirable properties in existing state of the art methods such as unbiased exact log-likelihood maximization, flexibility in density estimation and continuous-time. We train our model with default parameters on all datasets and it is able to match or outperform existing state of the art methods. We believe this model will ease the burden on practitioners for fitting new datasets.

References

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Remco R Bouckaert and Eibe Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 3–12. Springer, 2004.

Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

John Scott Butler. The statistical bias of numerically integrated statistical procedures. *Computers & mathematics with applications*, 11(6):587–593, 1985.

Patrick Ferdinand Christ, Florian Ettlinger, Georgios Kaissis, Sebastian Schlecht, Freba Ahmaddy, Felix Grün, Alexander Valentinitsch, Seyed-Ahmad Ahmadi, Rickmer Braren, and Bjoern Menze. Survivalnet: Predicting patient survival from diffusion weighted magnetic resonance images using cascaded fully convolutional and 3d convolutional neural networks. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 839–843. IEEE, 2017.

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

Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

Dominic Danks and Christopher Yau. Derivative-based neural modelling of cumulative distribution functions for survival analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 7240–7256. PMLR, 2022.

Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, Terry M Therneau, Colin L Colby, Raynell J Clark, Graham P Mead, Shaji Kumar, L Joseph Melton III, et al. Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, volume 87, pages 517–523. Elsevier, 2012.

David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in medicine*, 14(1):73–82, 1995.

Cheryl L Faucett, Nathaniel Schenker, and Jeremy MG Taylor. Survival analysis using auxiliary variables via multiple imputation, with application to aids clinical trial data. *Biometrics*, 58(1):37–47, 2002.

Thomas R Fleming and DY Lin. Survival analysis in clinical trials: past developments and future directions. *Biometrics*, 56(4):971–983, 2000.

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.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *J. Mach. Learn. Res.*, 21(85):1–63, 2020.

Xintian Han, Mark Goldstein, and Rajesh Ranganath. Survival mixture density networks. *arXiv preprint arXiv:2208.10759*, 2022.

Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

Hung Hung and Chin-Tsang Chiang. Estimation methods for time-dependent auc models with survival data. *Canadian Journal of Statistics*, 38(1):8–26, 2010.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

Vincent Jeanselme, Brian Tom, and Jessica Barrett. Neural survival clustering: Non-parametric mixture of neural networks for survival clustering. In *Conference on Health, Inference, and Learning*, pages 92–102. PMLR, 2022.

Adina Najwa Kamarudin, Trevor Cox, and Ruwanthi Kolamunnage-Dona. Time-dependent roc curve analysis in medical research: current methods and applications. *BMC medical research methodology*, 17(1):1–19, 2017.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

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): 1–12, 2018.

William A Knaus, Frank E Harrell, Joanne Lynn, Lee Goldman, Russell S Phillips, Alfred F Connors, Neal V Dawson, William J Fulkerson, Robert M Califf, Norman Desbiens, et al. The support prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of internal medicine*, 122(3):191–203, 1995.

Håvard Kvamme. havakv/pycox: Survival analysis with pytorch. https://github.com/havakv/pycox, 11 2022.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 2019.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Changhee Lee, Jinsung Yoon, and Mihaela Van Der Schaar. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 67(1):122–133, 2019a.

Changhee Lee, William Zame, Ahmed Alaa, and Mihaela Schaar. Temporal quilting for survival analysis. In *The 22nd international conference on artificial intelligence and statistics*, pages 596–605. PMLR, 2019b.

Yang Li, Dongzuo Liang, Shuangge Ma, and Chenjin Ma. Spatio-temporally smoothed deep survival neural network. *Journal of Biomedical Informatics*, 137:104255, 2023.

Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019.

L Mariani, D Coradini, E Biganzoli, P Boracchi, E Marubini, S Pilotti, B Salvadori, R Silvestrini, U Veronesi, R Zucali, et al. Prognostic factors for metachronous contralateral breast cancer: a comparison of the linear cox regression model and its artificial neural network extension. *Breast cancer research and treatment*, 44(2):167–178, 1997.

Xenia Miscouridou, Adler Perotte, Noémie Elhadad, and Rajesh Ranganath. Deep survival analysis: Nonparametrics and missingness. In *Machine Learning for Healthcare Conference*, pages 244–256. PMLR, 2018.

Chirag Nagpal, Rohan Sangave, Amit Chahar, Parth Shah, Artur Dubrawski, and Bhiksha Raj. Nonlinear semi-parametric models for survival analysis. *Proceedings of Machine Learning Research*, 2019.

Chirag Nagpal, Vincent Jeanselme, and Artur Dubrawski. Deep parametric time-to-event regression with time-varying covariates. In *Survival Prediction-Algorithms, Challenges and Applications*, pages 184–193. PMLR, 2021a.

Chirag Nagpal, Xinyu Li, and Artur Dubrawski. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3163–3175, 2021b.

Chirag Nagpal, Steve Yadlowsky, Negar Rostamzadeh, and Katherine Heller. Deep cox mixtures for survival regression. In *Machine Learning for Healthcare Conference*, pages 674–708. PMLR, 2021c.

Chirag Nagpal, Mononito Goswami, Keith Dufendach, and Artur Dubrawski. Counterfactual phenotyping with censored time-to-events. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022a.

Chirag Nagpal, Willa Potosnak, and Artur Dubrawski. auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data. *arXiv preprint arXiv:2204.07276*, 2022b.

Bernard Pereira, Suet-Feung Chin, Oscar M Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature communications*, 7(1):1–16, 2016.

Adler Perotte, Rajesh Ranganath, Jamie S Hirsch, David Blei, and Noémie Elhadad. Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis. *Journal of the American Medical Informatics Association*, 22(4):872–880, 2015.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020. URL http://jmlr.org/papers/v21/20-729.html.

Rajesh Ranganath, Adler J Perotte, Noémie Elhadad, and David M Blei. The survival filter: Joint survival analysis with a latent time series. In *UAI*, pages 742–751, 2015.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning for Healthcare Conference*, pages 101–114. PMLR, 2016.

Ori Rosen and Martin Tanner. Mixtures of proportional hazards regression models. *Statistics in Medicine*, 18(9):1119–1131, 1999.

Daniel J Sargent. Comparison of artificial neural networks with other statistical approaches: results from medical data sets. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 91(S8):1636–1642, 2001.

Marcel Scharth and Robert Kohn. Particle efficient importance sampling. *Journal of Econometrics*, 190(1):133–147, 2016.

Emilie M Shireman, Douglas Steinley, and Michael J Brusco. Local optima in mixture modeling. *Multivariate behavioral research*, 51(4):466–481, 2016.

Michael Sloma, Fayeq Syed, Mohammedreza Nemati, and Kevin S Xu. Empirical comparison of continuous and discrete-time representations for survival prediction. In *Survival Prediction-Algorithms, Challenges and Applications*, pages 118–131. PMLR, 2021.

Weijing Tang, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu. Soden: A scalable continuous-time survival model through ordinary differential equation networks. *J. Mach. Learn. Res.*, 23:34–1, 2022.

Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

Antonio Viganò, Marlene Dorgan, Jeanette Buckingham, Eduardo Bruera, and Maria E Suarez-Almazor. Survival prediction in terminal cancer patients: a systematic review of the medical literature. *Palliative Medicine*, 14(5):363–374, 2000.

Zifeng Wang and Jimeng Sun. Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–9, 2022.

Anny Xiang, Pablo Lapuerta, Alex Ryutov, Jonathan Buckley, and Stanley Azen. Comparison of the performance of neural network methods and cox regression for censored survival data. *Computational statistics & data analysis*, 34(2):243–257, 2000.

Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547. IEEE, 2016.

Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7234–7242, 2017.

Hyperparameters

All models have an equal training length of 4000 epochs. We pick the best-performing model with respect to their validation loss. The hyper-parameter spaces of each benchmark model are listed below.

CoxPH.

'alpha': [0, 1e-3, 1e-2, 1e-1],

DeepSurv.

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],
'layers_': [2, 3],
'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],

RSF.

'max_depth' : [None, 5],
'n_estimators' : [50, 100, 150, 200, 150],
'max_features' : [50, 75, sqrt(d), d//2, d],
'min_samples_split' : [10, 150, 200, 250],

'max_depth':None means that the expansion continues until all leaves are pure.

DSM.

'k ': [3, 4, 6],
'distribution' : ['Weibull', 'LogNormal'],
'learning_rate' : [1e-4, 5e-4, 1e-3],
'nodes_' : [48, 64, 96, 256],
'hidden_layers_': [1, 2, 3],
'discount': [1/3, 3/4, 1],
'batch_size': [128, 256],

DCM.

'k' : [3, 4, 6],
'nodes_' : [48, 64, 96, 256],
'hidden_layers_': [1, 2, 3],
'batch_size': [128, 256],
'use_activation': [True, False],

Deep-Hit.

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],

'hidden_layers_': [2, 3],

'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],

'alpha': [1e-1, 2e-1, 4e-1, 8e-1, 1],

'sigma' : [1e-1, 2.5e-1, 4e-1, 8e-1, 1, 2, 10],

'num_durations' : [10, 50, 100],

Cox-Time.

'lr' : [5e-4, 1e-3],

'batch_size': [256, 512, 1024],

'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],

'nodes_':[128, 256, 512],

'hidden_layers_': [1,2],

'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],

'lambda': [0, 1e-3, 1e-2, 1e-1],

'log_duration' : [True, False],

Ours.

'lr' : 2e-3,

'batch_size': 256,

'imps_size': [64, 256, 512],

'architecture': ['A1', 'A2'],

'layer_norm' : True,

'weight_decay': 1e-5,

'nodes_': 400

'layers_': 2,

'dropout': 4e-1,

'act': selu,