

SODEN: A Scalable Continuous-Time Survival Model through Ordinary Differential Equation Networks

Weijing Tang*

*Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA*

WEIJTANG@UMICH.EDU

Jiaqi Ma*

*School of Information
University of Michigan
Ann Arbor, MI 48109, USA*

JIAQIMA@UMICH.EDU

Qiaozhu Mei

*School of Information and Department of EECS
University of Michigan
Ann Arbor, MI 48109, USA*

QMEI@UMICH.EDU

Ji Zhu

*Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA*

JIZHU@UMICH.EDU

Editor: Jie Peng

Abstract

In this paper, we propose a flexible model for survival analysis using neural networks along with scalable optimization algorithms. One key technical challenge for directly applying maximum likelihood estimation (MLE) to censored data is that evaluating the objective function and its gradients with respect to model parameters requires the calculation of integrals. To address this challenge, we recognize from a novel perspective that the MLE for censored data can be viewed as a differential-equation constrained optimization problem. Following this connection, we model the distribution of event time through an ordinary differential equation and utilize efficient ODE solvers and adjoint sensitivity analysis to numerically evaluate the likelihood and the gradients. Using this approach, we are able to 1) provide a broad family of continuous-time survival distributions without strong structural assumptions, 2) obtain powerful feature representations using neural networks, and 3) allow efficient estimation of the model in large-scale applications using stochastic gradient descent. Through both simulation studies and real-world data examples, we demonstrate the effectiveness of the proposed method in comparison to existing state-of-the-art deep learning survival analysis models. The implementation of the proposed SODEN approach has been made publicly available at <https://github.com/jiaqima/SODEN>.

Keywords: Survival Analysis, Ordinary Differential Equation, Neural Networks

*. Equal contribution.

1. Introduction

Survival analysis is an important branch of statistical learning where the outcome of interest is the time until occurrence of an event, such as survival time until death and lifetime of a device until failure. In real-world data collections, some events may not be observed due to a limited observation time window or missing follow-up, which is known as *censoring*. In this case, instead of observing an *event time*, we record a *censored time*, for example, the end of the observation window, to indicate that no event has occurred prior to it. Survival analysis methods take into account the partial information contained in the censored data and have crucial applications in various real-world problems, such as rehospitalization, cancer survival in healthcare, reliability of devices, and customer lifetime (Chen et al., 2009; Miller Jr., 2011; Modarres et al., 2016).

Modern data collections have been growing in both scale and diversity of formats. For example, electronic health records of millions of patients over several decades are readily available, and they include laboratory test results, radiology images, and doctors’ clinical notes. Work towards more flexible and scalable modeling of event times has attracted great attention in recent years. In particular, various deep neural network models have been introduced into survival analysis due to their ability in automatically extracting useful features from large-scale raw data (Faraggi and Simon, 1995; Ching et al., 2018; Katzman et al., 2018; Lee et al., 2018; Gensheimer and Narasimhan, 2019; Chapfuwa et al., 2018; Kvamme et al., 2019; Steingrimsson and Morrison, 2020; Zhao, 2021).

As a natural choice for estimating a probabilistic model, likelihood-based methods have been widely used for both traditional and deep survival analysis. However, a major challenge for scalable maximum likelihood estimation of neural network models lies in difficult-to-evaluate integrals due to the existence of censoring. Specifically, for an uncensored observation i whose event time $T = t_i$ is recorded, the likelihood is the *probability density function* (PDF) $p(t_i)$. But, for a censored observation j , only the censored time $C = t_j$ is recorded while the event time T is unknown. The likelihood of observation j is the *survival function* $S(t_j)$, which is the probability of no event occurring prior to t_j : $S(t_j) = \mathcal{P}\{T > t_j\} = 1 - \int_0^{t_j} p(s)ds$. This integral imposes an intrinsic difficulty for optimization: evaluating the likelihood and the gradient with respect to parameters requires the calculation of integrals, which usually has no closed forms for most flexible distribution families specified by neural networks.

To address this challenge, most existing works try to avoid the integrals in the following two ways: 1) making additional structural assumptions so that no integral is included in the objective function, such as partial-likelihood-based methods under the proportional hazard (PH) assumption (Cox, 1975), or making parametric assumption that leads to closed-form integration in the likelihood (Wei, 1992); 2) discretizing the continuous event time with pre-specified intervals so that the integral is simplified into a cumulative product. However, the structural and parametric assumptions are often restrictive and thus limit the flexibility of the model (Ng’andu, 1997; Zeng and Lin, 2007); further, stochastic gradient descent algorithms cannot be directly applied to the partial-likelihood-based objective functions and thus limit the scalability of the model. As for discretization of the event time, it will likely cause information loss and introduce pre-specified time intervals as hyper-parameters.

In this paper, we recognize that maximizing the likelihood function for censored data can be viewed as an optimization problem with differential equation (DE) constraints, and thereby tackle the aforementioned optimization challenges with an efficient numerical approach. We propose to specify the distribution of event time through an ordinary differential equation (ODE) and utilize well-implemented ODE solvers to numerically evaluate the likelihood and its gradients. In particular, we consider the *hazard function* $\lambda_x(\cdot)$ ¹ and its integral, the *cumulative hazard function* $\Lambda_x(\cdot)$, in an ODE with a fixed initial value:

$$\begin{cases} \Lambda'_x(t) = h(\Lambda_x(t), t; x, \theta) \\ \Lambda_x(0) = 0 \end{cases}, \quad (1)$$

where the function $h(\cdot, \cdot, \cdot, \theta)$ is modeled by a neural network taking the cumulative hazard $\Lambda_x(t)$, the time t , and the feature x as inputs and parameterized by θ . Since the likelihood given both uncensored and censored data can be re-written in a simple form of the hazard and the cumulative hazard¹, we can evaluate the likelihood function by solving the above ODE numerically. Moreover, the gradient of the likelihood with respect to θ can be efficiently calculated via adjoint sensitivity analysis, which is a general method for differentiating optimization objectives with DE constraints (Pontryagin et al., 1962; Plessix, 2006). We name the proposed method as SODEN, Survival model through Ordinary Differential Equation Networks.

In comparison to existing methods described above, the proposed SODEN is more flexible to handle event times allowing for a broad range of distributions without strong structural assumptions. Further, we directly learn a continuous-time survival model using an ODE network, which avoids potential information loss from discretizing event times. We empirically evaluate the effectiveness of SODEN through both simulation studies and experiments on real-world datasets, and demonstrate that SODEN outperforms state-of-the-art models in most scenarios.

The rest of the paper is organized as follows. In Section 2, we provide a brief background on survival analysis and related work. In Section 3, we describe the proposed model and the corresponding learning approach. We evaluate the proposed method using simulation studies in Section 4 and on real-world data examples in Section 5. Finally Section 6 concludes the paper.

2. Background

In this section, we provide necessary preliminaries on survival analysis and summarize existing related work.

2.1 Preliminaries

2.1.1 THE PROBABILISTIC FRAMEWORK OF SURVIVAL ANALYSIS

Denote the non-negative event time by T and the feature vector by X . We are interested in the conditional distribution of T given $X = x$. In addition to the PDF, the distribution of

1. The hazard function describes the instantaneous rate at which the event occurs given survival, and is a popular modeling target in survival analysis. Probabilistic meanings of the hazard, the cumulative hazard, and the likelihood form in terms of the hazard and cumulative hazard (see Eq. (2)) are shown in Section 2.1.

T can be uniquely determined by any one of the followings: the survival, the hazard, or the cumulative hazard function. We introduce definitions of these functions below. Denote the PDF by $p_x(\cdot)$ with $\int p_x(t) dt = 1$. The survival function $S_x(\cdot)$ is the probability that no event occurred before time t , that is $S_x(t) = \mathcal{P}\{T > t | X = x\}$. The hazard rate $\lambda_x(t)$ characterizes the instantaneous rate at which the event occurs for individuals that are surviving at time t , which is denoted by

$$\lambda_x(t) = \lim_{\epsilon \rightarrow 0} \frac{\mathcal{P}\{t < T \leq t + \epsilon | T > t, X = x\}}{\epsilon} = \frac{p_x(t)}{S_x(t)}.$$

The cumulative hazard function $\Lambda_x(\cdot)$ is the integral of the hazard, that is $\Lambda_x(t) = \int_0^t \lambda_x(u) du$. It follows that $S_x(t) = \exp(-\Lambda_x(t)) = \exp\left(-\int_0^t \lambda_x(u) du\right)$. Thus, either the hazard function or the cumulative hazard function can specify the distribution of T . In particular, the hazard function $\lambda_x(\cdot)$ is a popular modeling target due to its practical meaning and informativeness in survival analysis.

2.1.2 LIKELIHOOD FUNCTION

Below, we provide the likelihood for a family of distributions given independent identically distributed (i.i.d.) observations. We consider the common right-censoring scenario where the event time T can be observed only if it does not exceed the censoring time C . Let $Y = \min\{T, C\}$ indicate the observed time and $\Delta = 1_{\{T \leq C\}}$ indicate whether we observe the actual event time. We observe i.i.d. triplets $D_i = (y_i, \Delta_i, x_i)$ for $i = 1, \dots, N$. Under the standard conditional independence assumption of the event time and the censoring time given features, the likelihood function is proportional to

$$\prod_{i=1}^N p_{x_i}(y_i)^{\Delta_i} S_{x_i}(y_i)^{1-\Delta_i} = \prod_{i=1}^N \lambda_{x_i}(y_i)^{\Delta_i} e^{-\Lambda_{x_i}(y_i)}, \quad (2)$$

where uncensored observations contribute the PDF and censored observations contribute the survival function. By definition, the likelihood function can also be written in terms of the hazard and the cumulative hazard as in (2).

2.2 Related Work

2.2.1 TRADITIONAL SURVIVAL ANALYSIS

There has been a large body of classical statistical models dealing with censored data in the literature. The Cox model (Cox, 1972), which is probably the most commonly used model in survival analysis, makes the proportional hazard (PH) assumption where the ratio of the hazard function is constant over time. Specifically, the hazard function consists of two terms: an unspecified baseline hazard function and a relative risk function, that is

$$\lambda_x(t) = \lambda_0(t) \exp(g(x; \theta)). \quad (3)$$

The Cox model also assumes that the relative risk linearly depends on features, that is $g(x; \theta) = x^T \theta$. In practice, however, either or both of the above assumptions are often violated. As a consequence, many alternative models have been proposed (Aalen, 1980;

Buckley and James, 1979; Gray, 1994; Bennett, 1983; Cheng et al., 1995; Lin and Ying, 1995; Fine et al., 1998; Chen et al., 2002; Shen, 2000; Wu and Witten, 2019). Among them, to address the limitation of multiplicative hazard, a broader family that involves multiplicative and additive hazard rate has been proposed (Aalen, 1980; Lin and Ying, 1995). To address the limitation of time-invariant effects, Gray (1994) has adapted the Cox model with time-varying coefficients to capture temporal feature effects. Alternatively, the accelerated failure time (AFT) model assumes that the logarithm of the event time is linearly correlated with features (Buckley and James, 1979; Wei, 1992), that is $\log T = x^T \theta + \epsilon$. When the error ϵ follows a specific parametric distribution such as log-normal and log-logistic, the likelihood in (2) under AFT model has a closed-form and can be efficiently optimized. Although the aforementioned models are useful, they often model the effect of features on the survival distribution in a simple, if not linear, way. These restrictions prevent the traditional models from being flexible enough to model modern data with increasing complexity.

2.2.2 DEEP SURVIVAL ANALYSIS

There has been an increasing research interest on utilizing neural networks to improve feature representation in survival analysis. Earlier works (Faraggi and Simon, 1995; Ching et al., 2018; Katzman et al., 2018) adapted the Cox model to allow nonlinear dependence on features but still make the PH assumption. For example, Katzman et al. (2018) used neural networks to model the relative risk $g(x; \theta)$ in (3). Kvamme et al. (2019) further allowed the relative risk to vary with time, which resulted in a flexible model without the PH assumption. Specifically, they extended the relative risk as $g(t, x; \theta)$ to model interactions between features and time. These models are all trained by maximizing the partial likelihood (Cox, 1975) or its modified version, which does not need to compute the integrals included in the likelihood function. The partial likelihood function is given by

$$\mathcal{PL}(\theta; D) = \prod_{i: \Delta_i=1} \frac{\exp(g(y_i, x_i; \theta))}{\sum_{j \in R_i} \exp(g(y_i, x_j; \theta))}, \quad (4)$$

where $R_i = \{j : y_j \geq y_i\}$ denotes the set of individuals who survived longer than the i^{th} individual, which is known as the *at-risk* set. Note that evaluation of the partial likelihood for an uncensored observation requires access to all other observations in the at-risk set. Hence, stochastic gradient descent (SGD) algorithms cannot be directly applied to partial likelihood-based objective functions, which is a serious limitation in training deep neural networks for large-scale applications. In the worst case, the risk set can be as large as the full data set. When the PH assumption holds, i.e., the numerators and denominators in (4) do not depend on y_i , evaluating the partial likelihood has a time complexity of $O(N)$ by computing $g(x_i; \theta)$ once and storing the cumulative sums. For flexible non-PH models, under which the likelihood has the form as (4), the time complexity further increases to $O(N^2)$. Although in practice one can naively restrict the at-risk set within each mini-batch, there is a lack of theoretical justification for this ad-hoc approach and the corresponding objective function is unclear.

On the other hand, SGD-based algorithms can be naturally applied to the original likelihood function. Following this direction, Lee et al. (2018) and Gensheimer and Narasimhan (2019) propose to discretize the continuous event time with pre-specified intervals, such that

Model	Non-linear	No PH Assumption	Continuous-time	SGD
Cox	✗	✗	✓	? ¹
DeepSurv	✓	✗	✓	?
DeepHit	✓	✓	✗	✓
Nnet-survival	✓	✓	✗	✓
Cox-Time	✓	✓	✓	?
SODEN (proposed)	✓	✓	✓	✓

Table 1: Comparison between the proposed method, SODEN, and related work, Cox (Cox, 1972), DeepSurv (Katzman et al., 2018), DeepHit (Lee et al., 2018), Nnet-survival (Gensheimer and Narasimhan, 2019), and Cox-Time (Kvamme et al., 2019).

the integral in (2) is replaced by a cumulative product. This method scales well with large sample size and does not make strong structural assumptions. However, determining the break points for time intervals is non-trivial, since too many intervals may lead to unstable model estimation while too few intervals may cause information loss.

We note that there are works that also consider a continuous event time but they do not optimize the likelihood function. Instead, they target summary statistics of the event time distribution such as the restricted mean survival time or the survival probability at a fixed time point (Steingrimsson and Morrison, 2020; Zhao, 2021). During the review process, we became aware of an independent and concurrent related work (Groha et al., 2020), which proposes a neural-network-based ODE approach to model the Kolmogorov forward equation that characterizes the transition probabilities for multi-state survival analysis.

The proposed SODEN is a flexible continuous-time model and is trained by maximizing the likelihood function, where SGD-based algorithms can be applied. Table 1 summarizes the comparison between SODEN and several representative existing methods.

2.2.3 DE-CONSTRAINED OPTIMIZATION

DE-constrained optimization has wide and important applications in various areas, such as optimal control, inverse problems, and shape optimization (Antil and Leykekhman, 2018). One of the major contributions of this work is to recognize that the maximum likelihood estimation in survival analysis is essentially a DE-constrained optimization problem. Specifically, the maximum likelihood estimation (MLE) for the proposed SODEN can be rewritten as

$$\begin{aligned} & \max_{\theta} \sum_{i=1}^N \Delta_i \log h(\Lambda_{x_i}(y_i), y_i; x_i, \theta) - \Lambda_{x_i}(y_i) \\ & \text{subject to } \Lambda'_{x_i}(t) = h(\Lambda_{x_i}(t), t; x_i, \theta) \\ & \Lambda_{x_i}(0) = 0, \quad i = 1, \dots, N \end{aligned} \quad (5)$$

where the constraint is a DE parameterized by θ and the objective contains the solution of the DE. Therefore, maximizing the likelihood function (2) that contains the solution of

1. SGD algorithms for Cox, DeepSurv, and Cox-Time can be naively implemented in practice, but not theoretically justifiable due to the form of the objective functions.

the parameterized ODE can be viewed as an optimization problem with DE constraints as shown in (5)². By bringing the strength of existing DE-constrained optimization techniques, we are able to develop novel numerical approaches for MLE in survival analysis without compromising the flexibility of models. There has been a rich literature on evaluating the gradient of the objective function in the DE-constrained optimization problem (Peto and Peto, 1972; Cao et al., 2003; Alexe and Sandu, 2009; Gerdtz, 2011). Among them, the adjoint sensitivity analysis is computationally efficient when evaluating the gradient of a scalar function with respect to large number of model parameters (Cao et al., 2003). Therefore, we use the adjoint method to compute the gradient of (5), whose detailed derivation is provided in Section 3.2.1.

DE-constrained optimization has also found its applications in deep learning. Chen et al. (2018) and Dupont et al. (2019) recently used ODEs parameterized with neural networks to model continuous-depth neural networks, normalizing flows, and time series, which lead to DE-constrained optimization problems. In this work, we share the merits of parameterizing the ODEs with neural networks but study a novel application of DE-constrained optimization in survival analysis.

3. The Proposed Approach

3.1 Survival Model through ODE Networks

We consider the cumulative hazard function $\Lambda_x(\cdot)$ through an ODE (1) with a fixed initial value. For readers' convenience, we repeat it below:

$$\begin{cases} \Lambda'_x(t) = h(\Lambda_x(t), t; x, \theta) \\ \Lambda_x(0) = 0 \end{cases},$$

where the function h determines the dynamic change of $\Lambda_x(\cdot)$: the derivative of cumulative hazard at time t is determined by the current cumulative hazard $\Lambda_x(t)$, the current time t , and feature x through the function h parameterized by θ . The initial value implies that the event always occurs after time 0 since $S_x(0) = \exp(-\Lambda_x(0)) = 1$. Given an individual's feature vector x and the parameter vector θ , for any specific time point t^* , the cumulative hazard $\Lambda_x(t^*)$ can be obtained as the solution of the initial value problem (1) at the time t^* , and the hazard rate can be obtained as $\lambda_x(t^*) = h(\Lambda_x(t^*), t^*; x, \theta)$. Therefore, the function h fully determines the conditional distribution of the event time T as shown in Section 2.1. The existence and uniqueness of the solution can be guaranteed if h and its derivatives are Lipschitz continuous (Walter, 1998). In this paper, we specify h as a neural network and the above guarantees hold as long as the neural network has finite weights and Lipschitz non-linearities. In practice, we do not require the initial value problem (1) to have a closed-form solution. We can obtain $\Lambda_x(t^*)$ numerically using any ODE solver given the derivative function h , initial value at $t_0 = 0$, evaluating time $t_1 = t^*$, parameters θ , and features x , that is

$$\Lambda_x(t^*) = \text{ODESolver}(h, \Lambda_x(0) = 0, t_1 = t^*, x, \theta). \tag{6}$$

2. The optimization problem in (5) belongs to a subclass of DE-constrained optimization problems, with the generic form of $\min_{\theta} J(\Lambda, \theta)$, subject to $g_1(\Lambda(t), \Lambda'(t), t; \theta) = 0$ and $g_2(\Lambda(0); \theta) = 0$.

We consider a general ODE form, where $h(\cdot, t; x, \theta)$ is a feed-forward neural network taking $\Lambda_x(t)$, t , and x as inputs, and θ represents all parameters in the neural network. Specifically, the Softplus activation function (Dugas et al., 2001) is used to constrain the output of the neural network, i.e. the hazard function, to be always positive. We refer this general form as SODEN; note that SODEN is a flexible survival model as it does not make strong assumptions on the family of the underlying distribution or how features x affect the event time.

Remark 1. Although there are other modeling alternatives that can uniquely characterize the event distribution as mentioned in Section 2.1.1, we choose to model the hazard in ODE (1) for three reasons. First, the hazard function has been widely used as the modeling target for summarizing survival data in the literature, due to its meaningful interpretation and informativeness about the underlying mechanism of events (Klein and Moeschberger, 2003, Chapter 2). Next, the hazard function is easier to model compared to the survival function, in the sense that it requires fewer constraints for the neural network structure under the ODE framework. For example, if we replace the cumulative hazard $\Lambda_x(t)$ with $S_x(t)$ in ODE (1), we need to make sure the solution not only being monotonically decreasing in t but also being within $[0, 1]$ for any $t \geq 0$, which poses additional constraints on the structure of the neural network h . Last but not least, the hazard function itself is of direct interest in many applications. For example, recent works in operational planning requires knowledge of the hazard rate of the waiting time until the customer abandons the queue (Ibrahim and Whitt, 2009; Reed and Tezcan, 2012).

3.2 Model Learning

We optimize SODEN by maximizing the likelihood function (2) given i.i.d. observations. The negative log-likelihood function of the i^{th} observation can be written as

$$\mathcal{L}(\theta; D_i) \triangleq -\Delta_i \log h(\Lambda_{x_i}(y_i), y_i; x_i, \theta) + \Lambda_{x_i}(y_i), \quad (7)$$

where $\Lambda_{x_i}(y_i)$, as given in (6), also depends on parameters θ . Our goal is to minimize $\sum_{i=1}^N \mathcal{L}(\theta; D_i)$ with respect to θ .

For large-scale applications, we propose to use mini-batch SGD to optimize the criterion, where the gradient of \mathcal{L} with respect to θ is calculated through the adjoint method (Pontryagin et al., 1962). In comparison to naively applying the chain rule through all the operations used in computing the loss function, the adjoint method has the advantage of reducing memory usage and controlling numerical error explicitly in back-propagation.

Next, we demonstrate how the gradients can be obtained.

3.2.1 BACK-PROPAGATION THROUGH ADJOINT SENSITIVITY ANALYSIS

In the forward pass, we need to evaluate $\mathcal{L}(\theta; D_i)$ for each i in a batch. While there might be no closed form for the solution of (1), $\Lambda_{x_i}(y_i)$ can be numerically calculated using a black-box ODESolver in (6) and all other calculations are straightforward. In the backward pass, the only non-trivial part in the calculation of the gradients of \mathcal{L} with respect to θ is back-propagation through the black-box ODESolver in (6). We compute it by solving another augmented ODE introduced by adjoint sensitivity analysis. Specifically, let the adjoint $a(t)$

satisfy $a'(t) = -\frac{\partial h}{\partial \Lambda} a(t)$ with $a(y_i) = 1$, and then it follows that $\nabla_{\theta} \Lambda_{x_i}(y_i) = \int_0^{y_i} a \frac{\partial h}{\partial \theta} dt$. Therefore, the gradient can be obtained by evaluating the following augmented ODE

$$\begin{cases} s'(t) = [h(\Lambda(t), t; x_i, \theta), -a(t) \frac{\partial h}{\partial \Lambda}, -a(t) \frac{\partial h}{\partial \theta}] \\ s(y_i) = [\Lambda_{x_i}(y_i), 1, \mathbf{0}_{|\theta|}] \end{cases}, \quad (8)$$

with $s(t) = [\Lambda(t), a(t), \bar{s}(t)]$ at $t = 0$, i.e., $\nabla_{\theta} \Lambda_{x_i}(y_i) = \bar{s}(0)$. Note that this approach does not need to access internal operations of ODE solvers used in the forward pass. Moreover, modern ODE solvers allow one to control the trade-off between the computing time and accuracy. Also note that a GPU-based implementation of back-propagation following the above rule is available in the *torchdiffeq* library (Chen et al., 2018). We provide the detailed derivation of (8) in Appendix A for presentation integrity.

3.2.2 MINI-BATCHING WITH TIME-RESCALING TRICK

We also provide a practical time-rescaling trick for mini-batching to better exploit the existing GPU-based implementation of ODE solvers. Concatenating ODEs of different observations in a mini-batch into a single combined ODE system is a useful trick for efficiently solving multiple ODEs on GPU. However, the existing GPU-based ODE solvers and the adjoint method in Chen et al. (2018) require that all the individual ODEs share the same initial point t_0 and the evaluating point t_1 in the ODESolver (6), which is unfortunately not the case in SODEN. For the i^{th} observation in a mini-batch, the ODE (1) in the forward pass needs to be evaluated at the corresponding observed time $t_1 = y_i$. To mitigate this discrepancy, we propose a time-rescaling trick that allows us to get the solution of individual ODEs at different time points by evaluating the combined ODE at only one time point. The key observation is that we can align the evaluating points of individual ODEs by variable transformation. Let $H_i(t) = \Lambda_{x_i}(t \cdot y_i)$, for which the dynamics is determined by

$$\begin{cases} H_i'(t) = h(H_i(t), ty_i; x_i, \theta) y_i \triangleq \tilde{h}(H_i(t), t; (x_i, y_i), \theta) \\ H_i(0) = \Lambda_{x_i}(0 \cdot y_i) = 0 \end{cases}.$$

Since $H_i(1) = \Lambda_{x_i}(y_i)$ for all i , evaluating the combined ODE of all $H_i(s)$ at $s = 1$ once will give us the values of $\Lambda_{x_i}(y_i)$ for all i . We therefore can take advantage of the existing GPU-based implementation for mini-batching by solving the combined ODE system of $H_i(s)$ with the time-rescaling trick¹.

4. Simulation Study

In this section, we conduct a simulation study to illustrate that the proposed SODEN can fit well with data when the commonly used PH assumption does not hold. For ease of visualization, we consider events generated from two groups where their survival functions cross each other, thus the PH assumption is violated. Further, we also show the advantage of SODEN as a continuous-time model rather than a discrete-time model.

1. We note that some recently developed deep learning libraries (e.g., JAX (Bradbury et al., 2018)) could support mini-batching over complicated operations such as solving ODEs with different initial and evaluating time points without using the time-rescaling trick. However, the proposed rescaling trick provides an easy-to-implement extension for the *torchdiffeq* library and potentially other frameworks.

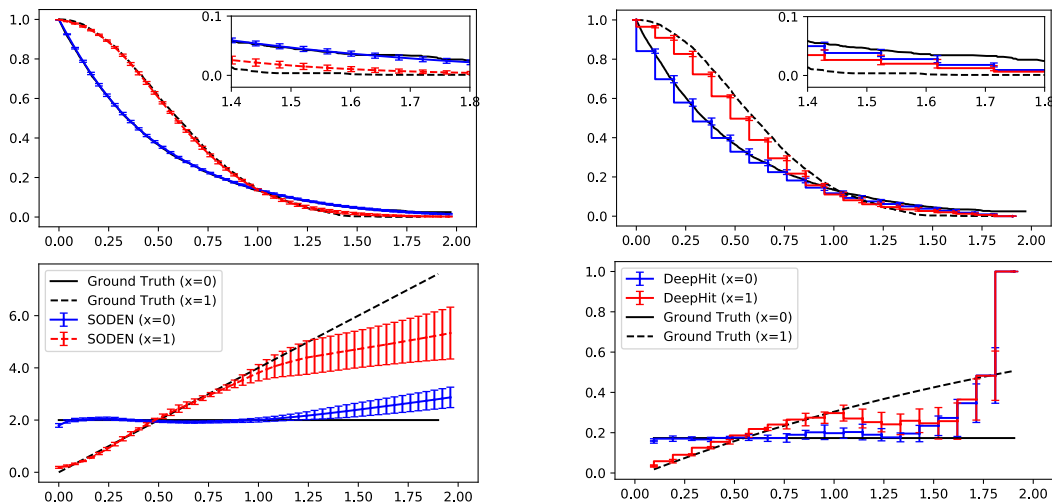


Figure 1: The survival functions (top row) and hazard functions (bottom row) of two groups, $x = 0$ and $x = 1$. The left column shows the results of SODEN, and the right column shows the results of DeepHit. In all figures, the results are the average of 10 independent trials and error bars indicate the standard deviation. The red curve indicates the predicted function for group $x = 1$ and the blue curve indicates the predicted function for group $x = 0$. The survival (Kaplan-Meier curves) and hazard functions corresponding to the data generating distribution for the two groups are shown in black curves (solid curves for group $x = 0$ and dashed curves for group $x = 1$).

4.1 Set-up

We generate event times from the conditional distribution defined by the survival function $S_x(t) = e^{-2t}I(x = 0) + e^{-2t^2}I(x = 1)$, where x follows a Bernoulli distribution with probability 0.5 and $I(\cdot)$ is the indicator function. The binary feature x can be viewed as an indicator for two groups of individuals. Note that the survival functions of the two groups, $S_0(t)$ and $S_1(t)$, cross at $t = 1$, hence the PH assumption does not hold. The censoring times were uniformly sampled between $(0, 2)$, which led to a censoring rate around 25%.

We apply the proposed SODEN and investigate the predicted survival functions and hazard functions under $x = 0$ and $x = 1$ respectively. We also provide the results of DeepHit (Lee et al., 2018), which is a discrete-time model without the PH assumption², to further illustrate the advantage of the continuous nature of SODEN. We train both models on the same simulated data with sample size 10,000. The reported results are based on 10 independent trials.

4.2 Results

The results of SODEN are shown in the left column of Figure 1. Note that the Kaplan-Meier (KM) estimate for each group can be considered a gold standard under our simulation setting, and we also plot them in Figure 1 as the true survival functions corresponding to the data

². See Appendix C for more details about this model.

generating distribution. The predicted survival functions generally agree well with the true survival functions (the upper-left figure). The predicted survival functions of the two groups cross approximately at $t = 1$, indicating SODEN can fit well with data not under the PH assumption. The lower-left figure shows that the predicted hazard functions of SODEN agree well with the true hazard functions when time is relatively small, but deviate from the true hazard functions as time increases. This is anticipated as there are few data points when t is large and there are many more data points when t is small. As a side note, while the estimate of the survival function looks better than that of the hazard function when t is large, it is a visual artifact. As the survival function is monotonically decreasing and bounded between 0 and 1, the deviation (as indicated by the error bar) of the estimated survival function from the ground truth near the tail is visually diminished. Relatively, the estimate of the survival function actually becomes worse for larger time t .

The results of DeepHit are shown in the right column of Figure 1. Due to the discrete nature of the model, both the survival functions and the hazard functions predicted by DeepHit are step functions. While the predicted survival functions (the upper-right figure) fit well with the true survival functions when t is small, the survival functions of the two groups are not well separated when t is large. As for the hazard function (the lower-right figure), similarly, the predicted hazard functions fit well when t is small but fluctuate wildly when t is large.

5. Real-world Examples

In this section, we demonstrate the effectiveness of SODEN by comparing it with five baseline models on three real-world datasets. We also conduct an ablation study to show the benefits of not making the PH assumption.

5.1 Datasets

We conduct experiments on the following three datasets: the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (**SUPPORT**), the Molecular Taxonomy of Breast Cancer International Consortium (**METABRIC**) (Katzman et al., 2018), and the Medical Information Mart for Intensive Care III (**MIMIC**) database (Johnson et al., 2016; Goldberger et al., 2000).

SUPPORT and METABRIC are two common survival analysis benchmark datasets, which have been used in many previous works (Katzman et al., 2018; Lee et al., 2018; Gensheimer and Narasimhan, 2019; Kvamme et al., 2019). We adopt the version pre-processed by Katzman et al. (2018) and refer readers there for more details. Despite their wide adoption in existing literature, we note that SUPPORT and METABRIC have relatively small sample sizes (8.8k for SUPPORT and 1.9k for METABRIC), which may not be ideal to evaluate deep survival analysis models.

In this paper, we further build a novel large-scale survival analysis benchmark dataset from the publicly available MIMIC database. The MIMIC database provides deidentified clinical data of patients admitted to an Intensive Care Unit (ICU) stay. We take adult patients who are alive 24 hours after the first admission to ICU. The event of interest is defined as the mortality after admission. The event time is observed if there is a record of death in the database; otherwise, the censored time is defined as the last time of being

Dataset	N	p	Censoring rate	Censored time (Yrs)		Observed time (Yrs)	
				Mean	Median	Mean	Median
MIMIC (MIMIC-SEQ)	35,304	26 (5+15×24)	61%	0.21	0.02	1.50	0.42
SUPPORT	8,873	14	32%	2.90	2.51	0.56	0.16
METABRIC	1,904	9	42%	0.44	0.43	0.27	0.24

Table 2: Summary statistics of three datasets. N is the sample size and p is the number of features. MIMIC-SEQ uses 5 time-static features and 15 time series features within the first 24 hours after admission.

discharged from the hospital. In MIMIC dataset, we extract 26 features based on the first 24-hour clinical data following Purushotham et al. (2018). In addition, to further evaluate deep learning models on applications with more complex data structure, we consider another feature set involving time series for the same group of patients, which is named as **MIMIC-SEQ** for differentiation. MIMIC-SEQ contains 5 time-static features and 15 time series features within the first 24 hours after admission. Following the protocols described above, we are able to get a dataset with over 35k samples.

The detailed summary statistics of the three datasets are provided in Table 2. In all datasets, the categorical features are encoded as dummy variables and all the features are standardized.

5.2 Models for Comparison

We compare the proposed method with the classical linear **Cox** model and four state-of-the-art neural-network-based models:

- **DeepSurv** is a PH model which replaces the linear feature combination in Cox with a neural network to improve feature extraction (Katzman et al., 2018).
- **Cox-Time** is a continuous-time model allowing non-PH, and is optimized by maximizing a modified partial-likelihood based loss function (Kvamme et al., 2019).
- **DeepHit** is a discrete-time survival model which estimates the probability mass at each pre-specified time interval, and is optimized by minimizing the linear combination of the negative log-likelihood and a differentiable surrogate ranking loss tailored for concordance index (Lee et al., 2018).
- **Nnet-Survival** also models discrete-time distribution via estimating the conditional hazard probability at each time interval (Gensheimer and Narasimhan, 2019).

Detailed model specifications and loss functions for the neural-network-based baselines can be found in Appendices B and C. Note that on the MIMIC-SEQ dataset, we only compare neural-network-based models.

In Section 4, we have shown that the proposed model, because of its flexible parameterization, is able to fit well to the simulated data where the PH assumption does not hold.

Here we further conduct an ablation study on real-world datasets to test the effect of the flexible parameterization. Specifically, we compare the general form of the proposed **SODEN**, with two of its degenerate variants, **SODEN-PH** and **SODEN-Cox**. SODEN-PH factorizes $h(\Lambda_x(t), t; x, \theta) = h_0(t; \theta)g(x; \theta)$ as a multiplication of two functions to satisfy the PH assumption, where both h_0 and g are specified as neural networks. SODEN-Cox is a linear version of SODEN-PH where $g(x) = e^{x\beta}$. Notably, SODEN-Cox and SODEN-PH are designed to have similar representation power as Cox and DeepSurv respectively.

5.3 Evaluation Metrics

Evaluating survival predictions needs to account for censoring. Here we describe several commonly used evaluation metrics (Kvamme et al., 2019; Wang et al., 2019).

5.3.1 TIME-DEPENDENT CONCORDANCE INDEX

Concordance index (C-index) (Harrell Jr. et al., 1984) is a commonly used discriminative evaluation metric in survival analysis, and it measures the probability that, for a random pair of observations, the relative order of the two event times is consistent with that of the two predicted survival probabilities. The C-index was originally designed for models using the PH assumption, where the relative order of the predicted survival probabilities for two given individuals does not change with time. Antolini et al. (2005) further propose time-dependent C-index for models without PH assumption, where the relative order of the predicted survival probabilities may be different if evaluated at different time points. In addition, Uno et al. (2011) introduce inverse probability weights to the C-index such that it does not depend on the study-specific censoring distribution. Following Antolini et al. (2005) and Uno et al. (2011), we adopt the inverse probability weighted time dependent C-index in our evaluation, which is given by

$$C^{td} = \frac{\sum_{i:\Delta_i=1} \sum_{j:y_i < y_j} I(\hat{S}_{x_i}(y_i) < \hat{S}_{x_j}(y_i)) / \hat{G}^2(y_i)}{\sum_{i:\Delta_i=1} \sum_{j:y_i < y_j} 1 / \hat{G}^2(y_i)},$$

where x_i , y_i , and Δ_i are the features, observed time, and event indicator for individual i ; $I(\cdot)$ is the indicator function; $\hat{S}_{x_i}(t)$ is the predicted survival function at time t given x_i ; and $\hat{G}(t)$ is the Kaplan-Meier estimator for the survival function of the censoring time, i.e. $\mathcal{P}(C > t)$. Under the independence assumption between the censoring time and the event time, C^{td} converges to the discrimination measure $\mathcal{P}(S_{x_i}(T_i) < S_{x_j}(T_i) | T_i < T_j)$.

In practice, the estimation of $\hat{G}(t)$ as well as the model predictions are relatively unstable for large t due to limited number of observations, yet they lead to large inverse probability weights $1/\hat{G}(t)$. Following Uno et al. (2011), we implement a truncated version of time-dependent C-index within a pre-specified time interval $(0, \tau)$, i.e.,

$$C_{\tau}^{td} = \frac{\sum_{i:\Delta_i=1, y_i < \tau} \sum_{j:y_i < y_j} I(\hat{S}_{x_i}(y_i) < \hat{S}_{x_j}(y_i)) / \hat{G}^2(y_i)}{\sum_{i:\Delta_i=1, y_i < \tau} \sum_{j:y_i < y_j} 1 / \hat{G}^2(y_i)}.$$

We report results under various τ with $\hat{G}(\tau) = 10^{-8}$, 0.2, and 0.4. When $\hat{G}(\tau) = 10^{-8}$, it is almost identical to the non-truncated version. Note that $C_{\tau}^{td}=1$ corresponds to a perfect ranking of predicted survival probabilities and $C_{\tau}^{td}=0.5$ corresponds to a random ordering.

5.3.2 INTEGRATED BRIER SCORE

For a binary classifier, the Brier score (BS) is defined as the mean square difference between the predicted probability and the ground-truth binary label. The metric BS can be decomposed into two components measuring calibration and discriminative performance respectively. Given similar discriminative performance, a lower BS indicates the closer the predicted survival probability $\hat{S}_x(t)$ is to the true probability of experiencing the event after time t . We refer well calibrated models to those with good probability estimates. Graf et al. (1999) generalized BS to take account for censoring in survival analysis. Specifically, the BS for survival analysis at time t is defined as

$$\text{BS}(t) = \frac{1}{N} \sum_{i=1}^N \left\{ \frac{(\hat{S}_{x_i}(t))^2 I(y_i \leq t, \Delta_i = 1)}{\hat{G}(y_i)} + \frac{(1 - \hat{S}_{x_i}(t))^2 I(y_i > t)}{\hat{G}(t)} \right\},$$

where the notations are the same as C^{td} . As the predicted survival probability depends on the time point of evaluation, we use integrated BS (IBS) to measure the overall BS on a time interval:

$$\text{IBS} = \frac{1}{t_{\max} - t_{\min}} \int_{t_{\min}}^{t_{\max}} \text{BS}(t) dt.$$

In practice, we choose the interval $[0, t_{\max}]$ with various t_{\max} satisfying $\hat{G}(t_{\max}) = 10^{-8}, 0.2$, and 0.4 , and compute this integral numerically by averaging over 100 grid points. The higher the IBS, the better the performance.

5.3.3 INTEGRATED BINOMIAL LOG-LIKELIHOOD

Graf et al. (1999) also generalized the binomial log-likelihood (BLL), which is a binary classification evaluation metric measuring both discrimination and calibration, to survival analysis in a similar way as BS. The BLL for survival analysis at time t is defined as

$$\text{BLL}(t) = \frac{1}{N} \sum_{i=1}^N \left\{ \frac{\log(1 - \hat{S}_{x_i}(t)) I(y_i \leq t, \Delta_i = 1)}{\hat{G}(y_i)} + \frac{\log(\hat{S}_{x_i}(t)) I(y_i > t)}{\hat{G}(t)} \right\},$$

where the notations are the same as BS. We can also define the integrated BLL (IBLL) to measure the overall performance from t_{\min} to t_{\max} , where

$$\text{IBLL} = \frac{1}{t_{\max} - t_{\min}} \int_{t_{\min}}^{t_{\max}} \text{BLL}(t) dt.$$

The higher the IBLL, the better the performance. Note that the IBS takes the squared error in the loss, i.e., error^2 , while the negative IBLL accounts for error with scale $-\log(1 - \text{error})$. Thus, in general, IBLL has larger magnitude than IBS and penalizes more for larger error.

5.3.4 NEGATIVE LOG-LIKELIHOOD

Negative log-likelihood (NLL) corresponds to $\mathcal{L}(\theta; D_i)$ in (7) and predictive NLL on held out data measures the goodness-of-fit of the model to the observed data. However, NLL is only applicable to models that provide likelihood, and it is not comparable between discrete-time

and continuous-time models due to the difference in the likelihood definition. We use NLL to compare three variants of SODEN in the ablation study. The lower the NLL, the better the performance.

5.4 Experimental Setup

We randomly split each dataset into training, validation and testing sets with a ratio of 3:1:1. To make the evaluation more reliable, we take 5 independent random splits for MIMIC(-SEQ), 10 independent random splits for SUPPORT and METABRIC as their sizes are relatively small. For each split, we train the Cox model on the combination of training and validation sets. For neural-network-based models, we train each model on the training set, and apply early-stopping using the loss on the validation set with patience 10. The hyper-parameters of each model are tuned within each split through 100 independent trials using random search. We select the optimal hyper-parameter setting with the best score on the validation set. For continuous-time models, DeepSurv, Cox-Time, and SODEN, the validation score is set as the loss. For discrete-time models, DeepHit and Nnet-Survival, the loss functions (i.e., NLLs) across different pre-specified time intervals are not comparable so the validation score is set as C^{td} as was done in Kvamme et al. (2019).

For all neural networks, we use multilayer perceptrons (MLP) with ReLU activation in all layers except for the output layer. For SODEN, Softplus is used to constrain the output to be always positive; for DeepHit and Nnet-Survival, Softmax and Sigmoid are used respectively to return PMF and discrete hazard probability. For the MIMIC-SEQ dataset, we incorporate a one-layer Gated Recurrent Units (GRU) encoder into the model architecture of each deep survival model to learn feature representation from sequence data. We use the RMSProp (Tieleman and Hinton, 2012) optimizer and tune batch size, learning rate, weight decay, momentum, the number of layers, and the number of neurons in each layer. The search ranges for the aforementioned hyper-parameters are shared across all neural-network-based models on each dataset. Additionally, we tune batch normalization and dropout for all neural-network-based baseline models. For DeepHit and Nnet-Survival, we tune the number of pre-specified time intervals. We also smooth the predicted survival function by interpolation, which is an important post-processing step to improve the performance of these discrete-time models. The tuning ranges of hyper-parameters are listed in Appendix D.

5.5 Results

5.5.1 DISCRIMINATIVE AND CALIBRATION PERFORMANCE

The comparison of model performances on MIMIC-SEQ, MIMIC, SUPPORT, and METABRIC are respectively reported in Tables 3 to 6.

We first consider the C-index metric, which measures the discriminative performance. We observe that the proposed SODEN outperforms other continuous-time models (Cox, DeepSurv, and Cox-Time). The differences in C-index are significant on all datasets, except for those with large τ on MIMIC-SEQ and MIMIC. The gain of SODEN against DeepSurv and Cox-Time demonstrates the benefits of not making the PH assumption and having a principled likelihood objective. We also observe that all neural network models significantly outperform the Cox model in almost all cases.

$\mathcal{P}(C > \tau)$	Model	C_{τ}^{td} (\uparrow)	IBLL (\uparrow)	IBS (\downarrow)
10^{-8}	DeepSurv	$0.685 \pm .002$	$-0.335 \pm .003$	$0.103 \pm .001$
	Cox-Time	$0.681 \pm .002$	$-0.332 \pm .003$	$0.103 \pm .001$
	Nnet-Survival	$0.679 \pm .003^*$	<u>$-0.331 \pm .003^*$</u>	$0.104 \pm .001$
	DeepHit	$0.688 \pm .002$	$-0.336 \pm .005^*$	$0.106 \pm .001^*$
	SODEN (ours)	<u>$0.687 \pm .002$</u>	$-0.328 \pm .004$	$0.103 \pm .001$
0.2	DeepSurv	$0.685 \pm .002$	$-0.400 \pm .013$	$0.124 \pm .003$
	Cox-Time	$0.681 \pm .002$	$-0.397 \pm .011$	<u>$0.125 \pm .003$</u>
	Nnet-Survival	$0.679 \pm .003^*$	<u>$-0.396 \pm .011^*$</u>	$0.126 \pm .003$
	DeepHit	$0.688 \pm .002$	$-0.402 \pm .012^*$	$0.128 \pm .004^*$
	SODEN (ours)	<u>$0.687 \pm .002$</u>	$-0.391 \pm .011$	<u>$0.125 \pm .004$</u>
0.4	DeepSurv	$0.740 \pm .002^*$	$-0.386 \pm .013^*$	$0.121 \pm .005^*$
	Cox-Time	$0.744 \pm .003^*$	$-0.382 \pm .014^*$	<u>$0.120 \pm .005^*$</u>
	Nnet-Survival	$0.737 \pm .004^*$	$-0.391 \pm .015$	$0.123 \pm .006$
	DeepHit	$0.752 \pm .003$	<u>$-0.381 \pm .014$</u>	<u>$0.120 \pm .005^*$</u>
	SODEN (ours)	$0.752 \pm .003$	$-0.374 \pm .013$	$0.118 \pm .005$

Table 3: Comparison of time dependent concordance index (C_{τ}^{td}), integrated binomial log-likelihood (IBLL), integrated brier score (IBS) on MIMIC-SEQ. The **bold** and underline markers denote the best and the second best performance respectively. The (\pm) error bar denotes the standard error of the mean. The asterisk (*) after a baseline model performance indicates a significant (either positive or negative) difference between that baseline model and the proposed SODEN, under pairwise t-test with p-value < 0.05 .

For discrete-time models, Nnet-Survival and DeepHit show strong discriminative performance on the C-index metric compared to continuous-time models in general. This is not surprising due to the facts that 1) similar as SODEN, the discrete-time models do not make strong structural assumptions; 2) the discrete-time models are tuned with C-index as the validation metric, and DeepHit has an additional ranking loss tailored for C-index. However, we find their advantage diminishes on MIMIC-SEQ and MIMIC, where the data size is much larger. We suspect the information loss due to discretizing the event time becomes more severe as the data size grows, and will eventually turn to the discriminative performance bottleneck.

We then consider the IBLL and IBS metrics, which measure a combination of the discriminative performance and the calibration performance. Overall, most models are similarly well-calibrated. However, DeepHit is obviously less calibrated than most other models, given it has the worst IBLL and IBS and the best C-index metric in most settings. This may be due to the surrogate ranking loss used in DeepHit.

In summary, the proposed SODEN demonstrates significantly better discriminative performance than all continuous-time baseline methods on all datasets. On the larger datasets (MIMIC-SEQ and MIMIC), SODEN achieves better or similar C-index metric compared to the discrete models. The superior discriminative performance of DeepHit comes at the price of the inferior calibration performance.

$\mathcal{P}(C > \tau)$	Model	C_{τ}^{td} (\uparrow)	IBLL (\uparrow)	IBS (\downarrow)
10^{-8}	Cox	$0.660 \pm .001^*$	$-0.335 \pm .003^*$	$0.105 \pm .001^*$
	DeepSurv	$0.683 \pm .001$	$-0.326 \pm .005^*$	<u>$0.101 \pm .001^*$</u>
	Cox-Time	$0.680 \pm .001$	$-0.326 \pm .003^*$	<u>$0.101 \pm .001^*$</u>
	Nnet-Survival	$0.681 \pm .001$	<u>$-0.321 \pm .002$</u>	<u>$0.101 \pm .001$</u>
	DeepHit	$0.685 \pm .002$	$-0.327 \pm .003^*$	$0.102 \pm .001^*$
	SODEN (ours)	<u>$0.684 \pm .002$</u>	$-0.319 \pm .003$	$0.100 \pm .001$
0.2	Cox	$0.660 \pm .001^*$	$-0.413 \pm .007^*$	$0.132 \pm .003^*$
	DeepSurv	$0.683 \pm .001$	$-0.402 \pm .006^*$	<u>$0.127 \pm .002^*$</u>
	Cox-Time	$0.680 \pm .001$	$-0.404 \pm .007^*$	$0.128 \pm .002^*$
	Nnet-Survival	$0.682 \pm .001$	<u>$-0.398 \pm .007$</u>	<u>$0.127 \pm .003$</u>
	DeepHit	$0.685 \pm .002$	$-0.404 \pm .008^*$	$0.128 \pm .002^*$
	SODEN (ours)	<u>$0.684 \pm .002$</u>	$-0.395 \pm .006$	$0.126 \pm .002$
0.4	Cox	$0.706 \pm .003^*$	$-0.399 \pm .018^*$	$0.124 \pm .007^*$
	DeepSurv	$0.739 \pm .003^*$	$-0.387 \pm .016$	<u>$0.120 \pm .007^*$</u>
	Cox-Time	$0.737 \pm .003^*$	$-0.387 \pm .020^*$	<u>$0.120 \pm .007^*$</u>
	Nnet-Survival	$0.741 \pm .005$	<u>$-0.386 \pm .019^*$</u>	<u>$0.120 \pm .007^*$</u>
	DeepHit	$0.747 \pm .004$	$-0.404 \pm .023$	$0.128 \pm .009$
	SODEN (ours)	<u>$0.746 \pm .003$</u>	$-0.379 \pm .019$	$0.118 \pm .007$

Table 4: Comparison of performance on MIMIC. The notations share the same definitions as in Table 3.

Finally, we remark that the event time and censoring time in MIMIC both have heavily right-skewed distributions, as indicated by the large discrepancy between their mean and median in Table 2. On MIMIC and MIMIC-SEQ, including more testing data near the tail in evaluation ($\hat{G}(\tau) = 10^{-8}$ or 0.2^3) gives a worse C_{τ}^{td} compared to including less tail samples ($\hat{G}(\tau) = 0.4$). This is because models tend to have poor prediction performance near the tail due to limited number of observations, yet these tail samples get large inverse probability weights. This also explains why the differences in C_{τ}^{td} among different models are less significant when including more tail samples.

5.5.2 ABLATION STUDY

While the trend over Cox, DeepSurv, and SODEN has supported our conjecture that flexible parameterization by introducing non-linearity and not making the PH assumption is important for practical survival analysis on modern datasets, we further verify this conjecture by the ablation study with SODEN-PH and SODEN-Cox (see Table 7).

First, we observe that the relative differences in the C-index metric among SODEN-Cox, SODEN-PH, and SODEN are similar as those among Cox, DeepSurv, and SODEN. In fact,

3. On MIMIC and MIMIC-SEQ, both $\hat{G}(\tau) = 10^{-8}$ and $\hat{G}(\tau) = 0.2$ have a tiny number of samples being excluded due to the right-skewness of the censoring distribution, and thus are close to the non-truncated version C^{td} .

$\mathcal{P}(C > \tau)$	Model	C_{τ}^{td} (\uparrow)	IBLL (\uparrow)	IBS (\downarrow)
10^{-8}	Cox	$0.596 \pm .002^*$	$-0.568 \pm .001^*$	$0.194 \pm .001^*$
	DeepSurv	$0.609 \pm .003^*$	$-0.559 \pm .002^*$	$0.190 \pm .001^*$
	Cox-Time	$0.607 \pm .004^*$	$-0.565 \pm .002$	$0.191 \pm .001$
	Nnet-Survival	$0.624 \pm .003$	$-0.570 \pm .004$	$0.193 \pm .001^*$
	DeepHit	$0.631 \pm .003$	$-0.583 \pm .006^*$	$0.197 \pm .001^*$
	SODEN (ours)	<u>$0.627 \pm .003$</u>	<u>$-0.563 \pm .002$</u>	<u>$0.191 \pm .001$</u>
0.2	Cox	$0.596 \pm .002^*$	$-0.585 \pm .001^*$	$0.201 \pm .000^*$
	DeepSurv	$0.609 \pm .003^*$	$-0.577 \pm .002$	$0.197 \pm .001$
	Cox-Time	$0.606 \pm .004^*$	$-0.583 \pm .002$	$0.199 \pm .001$
	Nnet-Survival	$0.623 \pm .003$	$-0.586 \pm .003$	$0.201 \pm .001^*$
	DeepHit	$0.630 \pm .003$	$-0.601 \pm .006^*$	$0.205 \pm .002^*$
	SODEN (ours)	<u>$0.627 \pm .003$</u>	<u>$-0.579 \pm .002$</u>	<u>$0.198 \pm .001$</u>
0.4	Cox	$0.595 \pm .002^*$	$-0.602 \pm .001^*$	$0.208 \pm .001^*$
	DeepSurv	$0.608 \pm .002^*$	$-0.595 \pm .002$	$0.205 \pm .001$
	Cox-Time	$0.605 \pm .004^*$	$-0.601 \pm .002$	$0.207 \pm .001$
	Nnet-Survival	$0.623 \pm .003$	$-0.602 \pm .003$	$0.208 \pm .001^*$
	DeepHit	$0.630 \pm .003$	$-0.619 \pm .007^*$	$0.212 \pm .002^*$
	SODEN (ours)	<u>$0.626 \pm .003$</u>	<u>$-0.597 \pm .002$</u>	$0.205 \pm .001$

Table 5: Comparison of performance on SUPPORT. The notations share the same definitions as in Table 3.

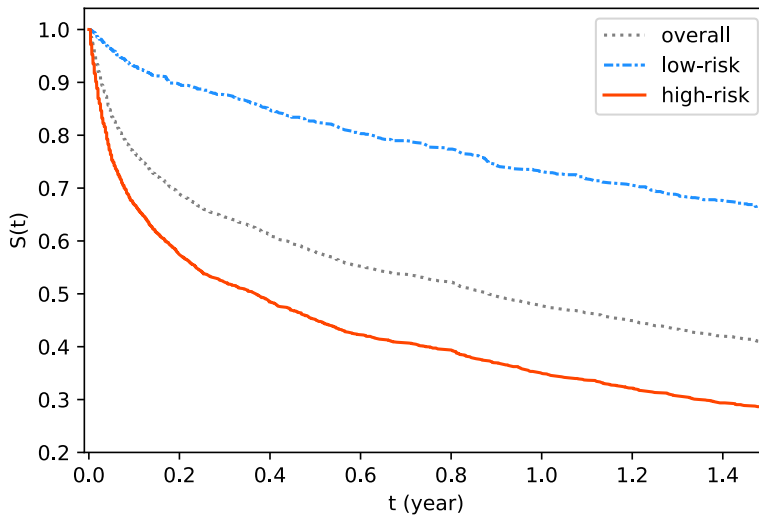


Figure 2: Kaplan-Meier curves of high/low-risk groups for SODEN on MIMIC.

we can see that the C_{τ}^{td} 's of SODEN-Cox and SODEN-PH in Table 7 are respectively similar with those of their partial-likelihood counterparts Cox and DeepSurv in Tables 3 to 6. This

$\mathcal{P}(C > \tau)$	Model	C_{τ}^{td} (\uparrow)	IBLL $_{\tau}$ (\uparrow)	IBS $_{\tau}$ (\downarrow)
10^{-8}	Cox	0.644 \pm .006*	<u>-0.508</u> \pm .009*	<u>0.169</u> \pm .002
	DeepSurv	0.635 \pm .007*	-0.517 \pm .011*	0.171 \pm .003*
	Cox-Time	0.648 \pm .007*	-0.511 \pm .009*	0.172 \pm .003*
	Nnet-Survival	<u>0.666</u> \pm .005	-0.510 \pm .007	0.171 \pm .002*
	DeepHit	0.674 \pm .006*	-0.514 \pm .004*	0.174 \pm .002*
	SODEN (ours)	0.661 \pm .005	-0.498 \pm .008	0.167 \pm .003
0.2	Cox	0.639 \pm .006*	<u>-0.521</u> \pm .006	<u>0.176</u> \pm .002
	DeepSurv	0.635 \pm .006*	-0.530 \pm .005*	0.179 \pm .002*
	Cox-Time	0.647 \pm .005*	-0.531 \pm .007*	0.179 \pm .002*
	Nnet-Survival	<u>0.662</u> \pm .004	-0.523 \pm .003	0.177 \pm .001
	DeepHit	0.671 \pm .004*	-0.533 \pm .003*	0.182 \pm .001*
	SODEN (ours)	0.659 \pm .003	-0.516 \pm .005	0.174 \pm .002
0.4	Cox	0.637 \pm .006*	-0.521 \pm .006	<u>0.175</u> \pm .002
	DeepSurv	0.635 \pm .006*	-0.526 \pm .005*	0.178 \pm .002*
	Cox-Time	0.644 \pm .005*	-0.526 \pm .006*	0.178 \pm .002*
	Nnet-Survival	<u>0.660</u> \pm .003	<u>-0.519</u> \pm .003	0.176 \pm .001
	DeepHit	0.668 \pm .004*	-0.528 \pm .003*	0.180 \pm .001*
	SODEN (ours)	0.658 \pm .004	-0.513 \pm .005	0.173 \pm .002

Table 6: Comparison of performance on METABRIC. The notations share the same definitions as in Table 3.

observation implies that 1) neural networks can approximate the baseline hazard function as well as the non-parametric Breslow’s estimator (Lin, 2007); 2) maximizing the likelihood function with numerical approximation approaches, where SGD based algorithms can be naturally applied, can perform as well as maximizing the partial likelihood for PH models.

Second, SODEN outperforms SODEN-PH and SODEN-Cox in terms of NLL by a large margin. The major difference between SODEN-PH and SODEN is that the former is restricted by the PH assumption while the latter is not. The comparison of NLL between SODEN-PH and SODEN provides a strong evidence that the PH assumption may not hold on these datasets. Further, SODEN-Cox often being the worst verifies again that both non-linearity and the flexibility of non-PH models matter.

5.5.3 RISK DISCRIMINATING VISUALIZATION

We further provide visualization of risk discrimination. We show the Kaplan-Meier curves (Kaplan and Meier, 1958) of high-risk and low-risk groups identified by SODEN on the MIMIC dataset. We first obtain the predicted survival probability for each individual at the median of all observed survival times in the test set. We then split the test set into high-risk and low-risk groups evenly based on their predicted survival probabilities. The Kaplan-Meier curves for the high-risk group, the low-risk group, and the entire test set are shown in Figure 2. The difference between high-risk and low-risk groups is statistically significant where the p-value of the log rank test (Peto and Peto, 1972) is smaller than 0.001.

Dataset	Metric ($\mathcal{P}(C > \tau)$)	SODEN	SODEN-PH	SODEN-Cox
MIMIC-SEQ	NLL	0.489 \pm .072	<u>0.520</u> \pm .069*	N/A
	C^{td} (10^{-8})	0.687 \pm .002	<u>0.682</u> \pm .001*	N/A
	C^{td} (0.2)	0.687 \pm .002	<u>0.683</u> \pm .001*	N/A
	C^{td} (0.4)	0.752 \pm .003	<u>0.739</u> \pm .005*	N/A
MIMIC	NLL	0.411 \pm .007	<u>0.436</u> \pm .007*	0.450 \pm .006*
	C^{td} (10^{-8})	0.684 \pm .002	<u>0.679</u> \pm .002*	0.659 \pm .001*
	C^{td} (0.2)	0.684 \pm .002	<u>0.679</u> \pm .002*	0.659 \pm .001*
	C^{td} (0.4)	0.746 \pm .003	<u>0.734</u> \pm .003*	0.706 \pm .003*
SUPPORT	NLL	0.676 \pm .008	<u>0.702</u> \pm .008*	0.761 \pm .022*
	C^{td} (10^{-8})	0.627 \pm .003	<u>0.608</u> \pm .003*	0.591 \pm .003*
	C^{td} (0.2)	0.627 \pm .003	<u>0.608</u> \pm .002*	0.590 \pm .004*
	C^{td} (0.4)	0.626 \pm .003	<u>0.607</u> \pm .002*	0.589 \pm .004*
METABRIC	NLL	0.149 \pm .015	0.176 \pm .013*	<u>0.167</u> \pm .010*
	C^{td} (10^{-8})	0.661 \pm .005	0.640 \pm .005*	<u>0.642</u> \pm .006*
	C^{td} (0.2)	0.659 \pm .003	<u>0.639</u> \pm .004*	0.638 \pm .005*
	C^{td} (0.4)	0.658 \pm .004	<u>0.639</u> \pm .005*	0.636 \pm .006*

Table 7: Comparison of negative log-likelihood (NLL) and time dependent concordance index (C_{τ}^{td}) between SODEN and its degenerate variants, SODEN-Cox and SODEN-PH, for ablation study. The **bold**, underline, and (\pm) error bar share the same definitions as in Table 3. The asterisk (*) indicates a significant difference between the proposed SODEN and its degenerate variants, under pairwise t-test with p-value < 0.05 .

6. Conclusion

In this paper, we have proposed a survival model through ordinary differential equation networks. It can model a broad range family of continuous event time distributions without strong structural assumptions and can obtain powerful feature representations using neural networks. Moreover, we have tackled the challenge of evaluating the likelihood of survival models and the gradients with respect to model parameters by an efficient numerical approach. The algorithm scales well by allowing direct use of mini-batch SGD. We have also demonstrated the effectiveness of the proposed method on both simulation studies and real-world data examples.

Acknowledgments

We would like to thank Yuanhao Liu and Chenkai Sun for running part of the baseline experiments; Yuekai Sun, Xuefei Zhang, and Debarghya Mukherjee for their feedback on the draft.

References

- Odd Aalen. A model for nonparametric regression analysis of counting processes. In *Mathematical Statistics and Probability Theory*, pages 1–25. Springer, New York, NY, 1980. ISBN 978-0-387-90493-1. doi: 10.1007/978-1-4615-7397-5_1.
- Mihai Alexe and Adrian Sandu. Forward and adjoint sensitivity analysis with continuous explicit Runge–Kutta schemes. *Applied Mathematics and Computation*, 208(2):328–346, 2009. ISSN 0096-3003. doi: <https://doi.org/10.1016/j.amc.2008.11.035>. URL <http://www.sciencedirect.com/science/article/pii/S0096300308008576>.
- Harbir Antil and Dmitriy Leykekhman. A brief introduction to PDE-constrained optimization. In *Frontiers in PDE-Constrained Optimization*, pages 3–40. Springer New York, New York, NY, 2018. ISBN 978-1-4939-8636-1. doi: 10.1007/978-1-4939-8636-1_1. URL https://doi.org/10.1007/978-1-4939-8636-1_1.
- Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24(24):3927–3944, 2005. doi: 10.1002/sim.2427. URL <https://doi.org/10.1002/sim.2427>.
- Steve Bennett. Analysis of survival data by the proportional odds model. *Statistics in Medicine*, 2(2):273–277, 1983. doi: 10.1002/sim.4780020223. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780020223>.
- James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- Jonathan Buckley and Ian James. Linear regression with censored data. *Biometrika*, 66(3): 429–436, 1979. ISSN 00063444. URL <http://www.jstor.org/stable/2335161>.
- Yang Cao, Shengtai Li, Linda Petzold, and Radu Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003. doi: 10.1137/S1064827501380630. URL <https://doi.org/10.1137/S1064827501380630>.
- Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Courtney Page, Benjamin Goldstein, Lawrence Carin, and Ricardo Henao. Adversarial time-to-event modeling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 735–744. 2018.
- Kani Chen, Zhezhen Jin, and Zhiliang Ying. Semiparametric analysis of transformation models with censored data. *Biometrika*, 89(3):659–668, 2002. ISSN 0006-3444. doi: 10.1093/biomet/89.3.659. URL <https://doi.org/10.1093/biomet/89.3.659>.
- Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*, pages 6571–6583. 2018.

- Yun Chen, Huirong Zhang, and Ping Zhu. Study of customer lifetime value model based on survival-analysis methods. In *2009 WRI World Congress on Computer Science and Information Engineering*, pages 266–270. 2009. doi: 10.1109/CSIE.2009.313.
- S. C. Cheng, Lee J. Wei, and Zhiliang Ying. Analysis of transformation models with censored data. *Biometrika*, 82(4):835–845, 1995. ISSN 0006-3444. doi: 10.1093/biomet/82.4.835. URL <https://doi.org/10.1093/biomet/82.4.835>.
- Travers Ching, Xun Zhu, and Lana X. Garmire. Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. *PLOS Computational Biology*, 14(4):e1006076, 2018. doi: 10.1371/journal.pcbi.1006076. URL <https://doi.org/10.1371/journal.pcbi.1006076>.
- David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 34(2):187–220, 1972. ISSN 00359246. URL <http://www.jstor.org/stable/2985181>.
- David R Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975. ISSN 00063444. URL <http://www.jstor.org/stable/2335362>.
- Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems 13*, pages 472–478. 2001.
- Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems 32*, pages 3140–3150. 2019.
- David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in Medicine*, 14(1):73–82, 1995. doi: 10.1002/sim.4780140108. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780140108>.
- Jason P. Fine, Zhiliang Ying, and Lee J. Wei. On the linear transformation model for censored data. *Biometrika*, 85(4):980–986, 1998. ISSN 0006-3444. doi: 10.1093/biomet/85.4.980. URL <https://doi.org/10.1093/biomet/85.4.980>.
- Michael F. Gensheimer and Balasubramanian Narasimhan. A simple discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019. doi: 10.7717/peerj.6257. URL <https://pubmed.ncbi.nlm.nih.gov/30701130>.
- Matthias Gerds. *Optimal Control of ODEs and DAEs*. De Gruyter, Berlin, Boston, 2011. ISBN 978-3-11-024999-6. doi: <https://doi.org/10.1515/9783110249996>. URL <https://www.degruyter.com/view/title/112116>.
- Ary L. Goldberger, Luis AN. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, Physiokit, and Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.cir.101.23.e215.

- 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. doi: 10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5. URL [https://doi.org/10.1002/\(SICI\)1097-0258\(19990915/30\)18:17/18<2529::AID-SIM274>3.0.CO;2-5](https://doi.org/10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5).
- Robert J. Gray. Spline-based tests in survival analysis. *Biometrics*, 50(3):640, 1994. ISSN 0006-341X. doi: 10.2307/2532779. URL <http://dx.doi.org/10.2307/2532779>.
- Stefan Groha, Sebastian M Schmon, and Alexander Gusev. Neural ODEs for multi-state survival analysis. *arXiv preprint arXiv:2006.04893*, 2020.
- Frank E. Harrell Jr., Kerry L. Lee, Robert M. Califf, David B. Pryor, and Robert A. Rosati. Regression modeling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2):143–152, 1984. doi: 10.1002/sim.4780030207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780030207>.
- Rouba Ibrahim and Ward Whitt. Real-time delay estimation in overloaded multiserver queues with abandonments. *Management Science*, 55(10):1729–1742, 2009. doi: 10.1287/mnsc.1090.1041.
- Alistair EW. Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016. doi: 10.1038/sdata.2016.35.
- Edward L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958. ISSN 01621459. doi: 10.2307/2281868. URL <http://www.jstor.org/stable/2281868>.
- 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. doi: 10.1186/s12874-018-0482-1. URL <https://doi.org/10.1186/s12874-018-0482-1>.
- John P. Klein and Melvin L. Moeschberger. *Basic Quantities and Models*, pages 21–61. Springer New York, New York, NY, 2003. ISBN 978-0-387-21645-4. doi: 10.1007/0-387-21645-6_2.
- 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. URL <http://jmlr.org/papers/v20/18-424.html>.
- Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 2314–2321. 2018. URL <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16160>.
- D. Y. Lin. On the Breslow estimator. *Lifetime Data Analysis*, 13(4):471–480, 2007. doi: 10.1007/s10985-007-9048-y. URL <https://doi.org/10.1007/s10985-007-9048-y>.

- D. Y. Lin and Zhiliang Ying. Semiparametric analysis of general additive-multiplicative hazard models for counting processes. *The Annals of Statistics*, 23(5):1712–1734, 1995. URL <https://www.jstor.org/stable/2242542>.
- Rupert G. Miller Jr. *Survival Analysis*. John Wiley & Sons, 2011.
- Mohammad Modarres, Mark P. Kaminskiy, and Vasilii Krivtsov. *Reliability Engineering and Risk Analysis: A Practical Guide*. CRC press, Boca Raton, 3rd edition, 2016. ISBN 9781315382425. doi: 10.1201/9781315382425. URL <https://doi.org/10.1201/9781315382425>.
- Nicholas H. Ng’andu. An empirical comparison of statistical tests for assessing the proportional hazards assumption of Cox’s model. *Statistics in Medicine*, 16(6):611–626, 1997. doi: 10.1002/(SICI)1097-0258(19970330)16:6<611::AID-SIM437>3.0.CO;2-T. URL [https://doi.org/10.1002/\(SICI\)1097-0258\(19970330\)16:6<611::AID-SIM437>3.0.CO;2-T](https://doi.org/10.1002/(SICI)1097-0258(19970330)16:6<611::AID-SIM437>3.0.CO;2-T).
- Richard Peto and Julian Peto. Asymptotically efficient rank invariant test procedures. *Journal of the Royal Statistical Society. Series A (General)*, 135(2):185–207, 1972. ISSN 00359238. doi: 10.2307/2344317. URL <http://www.jstor.org/stable/2344317>.
- R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006. doi: 10.1111/j.1365-246X.2006.02978.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-246X.2006.02978.x>.
- Lev Semenovich Pontryagin, EF Mishchenko, VG Boltyanskii, and RV Gamkrelidze. *Mathematical Theory of Optimal Processes*. Routledge, London, 1962. ISBN 9780203749319. doi: 10.1201/9780203749319. URL <https://doi.org/10.1201/9780203749319>.
- Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of Biomedical Informatics*, 83: 112 – 134, 2018. ISSN 1532-0464. doi: <https://doi.org/10.1016/j.jbi.2018.04.007>. URL <http://www.sciencedirect.com/science/article/pii/S1532046418300716>.
- Josh Reed and Tolga Tezcan. Hazard rate scaling of the abandonment distribution for the gi/m/n + gi queue in heavy traffic. *Operations Research*, 60(4):981–995, 2012. doi: 10.1287/opre.1120.1069.
- Xiaotong Shen. Linear regression with current status data. *Journal of the American Statistical Association*, 95(451):842–852, 2000. doi: 10.1080/01621459.2000.10474276. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.2000.10474276>.
- Jon Arni Steingrímsson and Samantha Morrison. Deep learning for survival outcomes. *Statistics in Medicine*, 39(17):2339–2349, 2020. doi: <https://doi.org/10.1002/sim.8542>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.8542>.
- Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural Networks for Machine Learning*, 4(2):26–31, 2012.

- 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.
- Wolfgang Walter. First order systems. Equations of higher order. In *Ordinary Differential Equations*, pages 105–157. Springer New York, New York, NY, 1998. ISBN 978-1-4612-0601-9. doi: 10.1007/978-1-4612-0601-9_4. URL https://doi.org/10.1007/978-1-4612-0601-9_4.
- Ping Wang, Yan Li, and Chandan K. Reddy. Machine learning for survival analysis: A survey. *Association for Computing Machinery (ACM) Computing Surveys*, 51(6), 2019. ISSN 0360-0300. doi: 10.1145/3214306. URL <https://doi.org/10.1145/3214306>.
- Lee J. Wei. The accelerated failure time model: A useful alternative to the Cox regression model in survival analysis. *Statistics in Medicine*, 11(14-15):1871–1879, 1992. doi: 10.1002/sim.4780111409. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780111409>.
- Jiacheng Wu and Daniela Witten. Flexible and interpretable models for survival data. *Journal of Computational and Graphical Statistics*, 28(4):954–966, 2019. doi: 10.1080/10618600.2019.1592758. URL <https://doi.org/10.1080/10618600.2019.1592758>.
- Donglin Zeng and D. Y. Lin. Maximum likelihood estimation in semiparametric regression models with censored data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):507–564, 2007. doi: 10.1111/j.1369-7412.2007.00606.x. URL <https://doi.org/10.1111/j.1369-7412.2007.00606.x>.
- Lili Zhao. Deep neural networks for predicting restricted mean survival times. *Bioinformatics*, 36(24):5672–5677, 01 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa1082. URL <https://doi.org/10.1093/bioinformatics/btaa1082>.

Appendix A. Derivation of Gradients through Adjoint Sensitivity Analysis

We rewrite $\Lambda_{x_i}(y_i)$ as the objective function $G(\Lambda, \theta) = \int_0^{y_i} h(\Lambda(t), t, x_i; \theta) dt$ with the following DE constraint

$$\begin{cases} \Lambda'(t) = h(\Lambda(t), t; x_i, \theta) \\ \Lambda(0) = 0 \end{cases}, \quad (9)$$

where we simplify the notation Λ_{x_i} as Λ . Now we wish to calculate the gradient of $G(\Lambda, \theta)$ with respect to θ subject to the DE constraint (9). Introducing a Lagrange multiplier $\xi(t)$, we form the Lagrangian function

$$I(\Lambda, \theta, \xi) = G(\Lambda, \theta) - \int_0^{y_i} \xi[\Lambda'(t) - h(\Lambda, t; x_i, \theta)] dt.$$

Because $\Lambda'(t) - h(\Lambda, t; x_i, \theta) = 0$ for any t , the gradient of G with respect to θ is equal to

$$\nabla_{\theta} G = \frac{\partial I}{\partial \theta} = \int_0^{y_i} (1 + \xi) \left(\frac{\partial h}{\partial \theta} + \frac{\partial h}{\partial \Lambda} \frac{\partial \Lambda}{\partial \theta} \right) dt - \int_0^{y_i} \xi \frac{\partial \Lambda'}{\partial \theta} dt.$$

Using integration by parts, it follows that

$$\begin{aligned} \nabla_{\theta} G &= \int_0^{y_i} (1 + \xi) \frac{\partial h}{\partial \theta} dt \\ &\quad + \int_0^{y_i} \frac{\partial \Lambda}{\partial \theta} \left[\xi' + (1 + \xi) \frac{\partial h}{\partial \Lambda} \right] dt - \left(\xi \frac{\partial \Lambda}{\partial \theta} \right) \Big|_0^{y_i}. \end{aligned}$$

Denote the adjoint $a(t) = \xi(t) + 1$ and let $a(t)$ satisfy $a'(t) = -\frac{\partial h}{\partial \Lambda} a(t)$ and $a(y_i) = 1$, then it follows that $\nabla_{\theta} G = \int_0^{y_i} a \frac{\partial h}{\partial \theta} dt$. Calculation of the above integral requires the value of $\Lambda(t)$ and $a(t)$ along their entire trajectory from 0 to y_i . Thus, we can compute the gradient $\nabla_{\theta} G$ by solving the following augmented ODE which concatenates the dynamics and initial states of the three. Specifically, let $s(t) = [\Lambda(t), a(t), \nabla_{\theta} G]$, then s follows the ODE in (8).

Appendix B. Partial Likelihood Based Methods

The Cox model (Cox, 1972) and its extensions such as DeepSurv (Katzman et al., 2018) and Cox-Time (Kvamme et al., 2019) consider the hazard function in a semi-parametric way. Specifically, the conditional hazard function is factorized into two terms: a non-parametric baseline hazard function and a parametric relative risk function, that is

$$\lambda_x(t) = h_0(t) \exp(g(t, x; \theta)).$$

The **Cox** model assumes a time-invariant linear relative risk where $g(t, x; \theta) = x^T \theta$. Subsequently, **DeepSurv** allows the relative risk to be a nonlinear function of feature x , i.e. $g(t, x; \theta) = g(x; \theta)$, but the proportional hazard assumption still holds; **Cox-Time** further allows the relative risk function to depend on time, which can handle the non-proportional hazard. In particular, DeepSurv and Cox-Time use neural networks to model $g(x; \theta)$ and $g(t, x; \theta)$.

All the above models are fitted in two steps: the parameters in the relative risk function are learned through maximizing the partial likelihood function (Cox, 1975); the non-parametric cumulative baseline hazard function is obtained through the Breslow's estimator (Lin, 2007) given the fitted relative risk in the first step.

Partial likelihood. The partial likelihood function is defined as

$$\mathcal{PL}(\theta; D) = \prod_{i:\Delta_i=1} \frac{\exp(g(y_i, x_i; \theta))}{\sum_{j \in R_i} \exp(g(y_i, x_j; \theta))},$$

where $R_i = \{j : y_j \geq y_i\}$ denotes the set of individuals who survived longer than the i -th individual, which is the so called *at-risk* set. The estimator of θ is obtained by minimizing the negative log-partial likelihood function, that is

$$\min_{\theta} \sum_{i:\Delta_i=1} [-g(y_i, x_i; \theta) + \log \sum_{j \in R_i} \exp(g(y_i, x_j; \theta))].$$

The partial likelihood function of each individual requires the access to the data of all individuals in the at-risk set. Hence, stochastic gradient decent (SGD) based algorithms cannot be directly applied. Although we can naively sample a mini-batch and restrict the at-risk set to individuals who are included in the current mini-batch in practice, there is a lack of theoretical justification.

Breslow's estimator. In order to obtain the predicted survival function, we need to estimate the cumulative hazard function. For models with the proportional hazard assumption, the estimated cumulative hazard function can be written as

$$\hat{\Lambda}_x(t) = \int_0^t \hat{h}_0(s) ds \cdot \exp(g(x; \hat{\theta})) = \hat{H}_0(t) \exp(g(x; \hat{\theta})),$$

where \hat{H}_0 is the estimated cumulative baseline hazard function. The Breslow's estimator for H_0 is given by

$$\hat{H}_0(t) = \sum_{i:y_i \leq t} \frac{\Delta_i}{\sum_{j \in R_i} \exp(g(x_j; \hat{\theta}))}.$$

For Cox-Time with non-proportional hazard, the estimated cumulative hazard function is given by

$$\hat{\Lambda}_x(t) = \sum_{i:y_i \leq t} \frac{\Delta_i}{\sum_{j \in R_i} \exp(g(y_i, x_j; \hat{\theta}))} \exp(g(y_i, x; \hat{\theta})).$$

The survival function can then be estimated by $\hat{S}_x(t) = \exp(-\hat{\Lambda}_x(t))$.

Appendix C. Discrete-Time Methods

In the discrete-time setting, the range of possible values of the event time T is divided into a set of disjoint intervals through pre-specified break points $\{t_0 = 0, t_1, \dots, t_L\}$. Denote the intervals by $I_l = (t_{l-1}, t_l]$, $l = 1, \dots, L$. Suppose the probability of occurrence of the event in time interval I_l is $p_l(x) \geq 0$ with $\sum_{l=1}^L p_l(x) = 1$. The cumulative distribution F_l and survival functions S_l are, respectively

$$F_l(x) = \mathcal{P}\{T \leq t_l | X = x\} = \sum_{j=1}^l p_j(x), \quad S_l(x) = \mathcal{P}\{T > t_l | X = x\} = 1 - F_l = 1 - \sum_{j=1}^l p_j(x).$$

The conditional hazard probability $\lambda_l(x)$ is the probability that the event occurs in interval I_l conditional on the survival up to the beginning of I_l , which could also determine the survival function through

$$\lambda_l(x) = \mathcal{P}\{T \in I_l | T \geq t_{l-1}, X = x\} = \frac{p_l(x)}{S_{l-1}(x)}, \quad S_l(x) = \prod_{j=1}^l (1 - \lambda_j(x)).$$

Under the conditional independence assumption of the event time and the censoring time given features, the likelihood function is proportional to

$$\prod_i p_{l_i}(x_i)^{\Delta_i} (1 - \sum_{j=1}^{l_i-1} p_j(x_i))^{1-\Delta_i} = \prod_i [\lambda_{l_i}(x_i) \prod_{j=1}^{l_i-1} (1 - \lambda_j(x_i))],$$

where l_i is the index of time interval satisfying $t_{l_i-1} < y_i \leq t_{l_i}$.

DeepHit (Lee et al., 2018) models the probability mass function where the output of the neural network is a vector $[p_1(x), \dots, p_L(x)]$. In addition to the negative log-likelihood (NLL) loss function, DeepHit considers another differentiable surrogate ranking loss tailored for time dependent concordance index, that is

$$\mathcal{L}_2 = \sum_{i:\Delta_i=1} \sum_{j:l_i < l_j} \eta(F_{l_i}(x_i), F_{l_j}(x_j)),$$

where $\eta(x, y) = \exp\left(\frac{-(x-y)}{\sigma}\right)$ and σ is a hyperparameter. They introduce another hyperparameter α to control the trade-off between the ranking loss and the NLL loss. **Nnet-Survival** (Gensheimer and Narasimhan, 2019) models the conditional hazard probability where the output of the neural network is a vector $[\lambda_1(x), \dots, \lambda_L(x)]$, and it is learned by maximizing the likelihood function.

Appendix D. Hyperparameter Tuning

We list the tuning ranges of hyperparameters for all neural network based models on three datasets in Table 8, where $\{\cdot\}$ represents the discrete search space and $[\cdot]$ represents the continuous search space⁴. Specifically, we tune the rate of dropout and batch normalization for DeepSurv, DeepHit, Nnet-Survival, and Cox-Time. We treat the number of time intervals as a hyperparameter for DeepHit and Nnet-Survival. We also tune two hyperparameters, α and σ , associated with the surrogate ranking loss in DeepHit. Since the three datasets are of different sizes, we use different search ranges for the batch size: $\{32, 64, 128, 256\}$ for METABRIC, $\{128, 256, 512\}$ for SUPPORT, and $\{512, 1024\}$ for MIMIC and MIMIC-SEQ. The discrete models (DeepHit and Nnet-Survival) appear to be sensitive to the number of time intervals on different datasets. Therefore we search the number of time intervals for these two discrete models from $\{10, 50, 100, 200, 400\}$ for the smaller datasets, METABRIC and SUPPORT, and from $\{50, 100, 200, 400, 800\}$ for the larger datasets, MIMIC and MIMIC-SEQ.

4. For the number of neurons, a real number is first sampled from the continuous space and then rounded to the closest integer.

Number of dense hidden layers	$\{1, 2, 4\}$
Number of neurons in each dense hidden layer	$[2^2, 2^7]$
Number of neurons in each GRU hidden layer	$[2^3, 2^8]$
Learning rate	$[10^{-4.5}, 10^{-1.5}]$
Weight decay	$[10^{-9}, 10^{-4}]$
Momentum	$[0.85, 0.99]$
Dropout (DeepHit, DeepSurv, Nnet-Survival, Cox-Time)	$\{0, 0.1, 0.5\}$
Batch normalization (DeepHit, DeepSurv, Nnet-Survival, Cox-Time)	$\{\text{True}, \text{False}\}$
α (Surrogate ranking loss in DeepHit)	$[0, 1]$
σ (Surrogate ranking loss in DeepHit)	$\{0.25, 1, 5\}$

Table 8: Tuning ranges of hyperparameters for experiments on the real-world datasets.