Survival Mixture Density Networks

Xintian Han XINTIAN.HAN@NYU.EDU
New York University

Mark Goldstein GOLDSTEIN@NYU.EDU
New York University

Rajesh Ranganath RAJESHR@CIMS.NYU.EDU
New York University

Abstract

Survival analysis, the art of time-to-event modeling, plays an important role in clinical treatment decisions. Recently, continuous time models built from neural ODEs have been proposed for survival analysis. However, the training of neural ODEs is slow due to the high computational complexity of neural ODE solvers. Here, we propose an efficient alternative for flexible continuous time models, called Survival Mixture Density Networks (Survival MDNs). Survival MDN applies an invertible positive function to the output of Mixture Density Networks (MDNs). While MDNs produce flexible real-valued distributions, the invertible positive function maps the model into the time-domain while preserving a tractable density. Using four datasets, we show that Survival MDN performs better than, or similarly to continuous and discrete time baselines on concordance, integrated Brier score and integrated binomial log-likelihood. Meanwhile, Survival MDNs are also faster than ODE-based models and circumvent binning issues in discrete models.

1. Introduction

Survival analysis serves as an important tool in healthcare to assess the risk of events, such as onset of disease (Wilson et al., 1998) or death (Pocock et al., 1982), rehospitalization (Patterson and Lee, 1998) and discharge from hospital (Wang et al., 2020). Survival modeling has been widely used in clinical applications, including improving the prognosis of cancer (Faradmal et al., 2012; Goldstraw et al., 2016; Wang et al., 2019; Lin and Anisa, 2021; Wang et al., 2021), predicting the onset of septic shock (Henry et al., 2015), assessing the survival time of heart failure patients (Ahmad et al., 2017; Kojoria et al., 2004; Jones et al., 2019; Yin et al., 2022) and estimating the graft survival rate of kidney transplant patients (Lee et al., 2019; Rodrigues et al., 2019).

Given patients' electronic health records including lab tests, vitals, radiology results and clinical notes, doctors need to determine a level of treatment based on the level of risk. For example, WHO guidelines suggest more aggressive treatments for higher risk cardiovascular disease patients (WHO et al., 2007). Therefore, an accurate model of risk is necessary.

Risk in survival analysis is characterized by the conditional distribution of the event time given a patient's healthcare records. What distinguishes survival analysis from traditional regression problems is that event times can be censored, i.e., only known to lie within a certain range. For example, patients may remain healthy throughout a 10-year coronary artery disease study (Wilson et al., 1998) so it is only known that such patients survive at least 10 years. Discarding censored times may introduce bias into estimates by underestimating the time until an event, because later times are more likely to be censored and thus thrown away.

Likelihood-based methods are used to estimate survival models (Kalbfleisch and Prentice, 2011). In addition to the usual mass or density computed in maximum likelihood problems, the survival likelihood for censored data includes the survival function, i.e., one minus the cumulative distribution function (CDF) of the distribution. For many distributions, CDF evaluations require explicitly integrating the density. Recent advances in deep learning provide opportunities for flexible survival modeling (LeCun et al., 2015; Ranganath et al., 2016). However, flexible distributions utilizing deep learning, such as those modeled by GANs (Goodfellow et al., 2014; Chapfuwa et al., 2018), may not yield efficient CDF computation.

To keep estimation tractable, traditional survival analysis techniques make distributional assumptions, e.g. log-normal density or proportional hazards (Kalbfleisch and Prentice, 2011; Cox, 1972). But this limits the flexibility of the model. To move beyond this, discrete time models divide continuous times into a sequence of bins (Miscouridou et al., 2018; Lee et al., 2018; Kvamme and Borgan, 2019) and can approximate arbitrary continuous distributions increasingly well as the number of bins increases. However, the choice of bin boundaries is troublesome: it is unclear how best to set the time intervals for each bin, and the survival function for times within a bin is ill-defined. ODE-based continuous time models (Tang et al., 2020, 2022) specify the time-to-event distribution through ODEs. However, the training of ODE-based models is slow due to expensive numerical integration requiring many neural network evaluations for each forward pass (Kelly et al., 2020).

In this work, we propose Survival Mixture Density Networks (Survival MDN). Survival MDN builds off mixture density networks (MDN) (Bishop, 1994) to allow flexible modeling. Since the time-to-event is positive in survival modeling, we apply an invertible positive function to the samples from MDNs. The CDF of Survival MDNs can be obtained easily through the evaluation of the CDF of the mixture components of the MDN, which is simple for mixture components like Gaussians. We evaluate Survival MDN and baselines on four clinical datasets: SUPPORT, METABRIC, GBSG, and MIMIC. On all datasets, Survival MDN performs better than, or as well as, the baselines on concordance, integrated Brier Score and integrated binomial log-likelihood. We also show that training Survival MDNs can be 100 times faster than the ODE-based model SODEN (Tang et al., 2020).[1]

Generalizable Insights about Machine Learning in the Context of Healthcare

The majority of flexible survival modeling relies on training with the Cox partial likelihood, discrete time modeling, or ordinary differential equations. Training with partial likelihood precludes the use of stochastic gradient descent and is not scalable for large datasets. Discrete time models have issues with choosing bin boundaries and determining the survival probability for a particular time. ODE-based models use likelihood for training but are slow to train. Our proposed model Survival MDN have several advantages 1) It is a continuous time model 2) It makes fewer distributional assumptions 3) It can be trained with stochas-

[1] The code is available at https://github.com/XintianHan/Survival-MDN

tic gradient descent 4) It is easier to use than discrete models and faster than ODE-based models.

2. Background

In this section, we introduce the mathematical foundation of survival analysis and summarize related works. We then describe how our work is distinguished from previous works.

2.1. Foundation of Survival Analysis

Survival analysis studies the distribution of event time T given covariates X. For example, we would like to know when a patient may die after the admission to ICU. The event time is called the failure time or survival time. We consider the common scenario of *right-censoring* in this work, where only a lower bound of the survival time is observed for some patients. We call the lower bound the *censored time* C. When $T > C$, only the censored time C is observed; when $T \leq C$,the failure time T is observed. We use $\Delta = I\{T \leq C\}$ to indicate whether the event time is observed and $U = \min\{T, C\}$ to denote the observed time.

A central quantity that appears in the estimation and use of survival models is the survival function $S(t|X) = P(T > t|X)$, i.e., the probability a patient with covariates X will survival until time t. By definition, $S(t|X) = 1 - \text{CDF}(t|X)$.

Assume we observe i.i.d. datapoints $\{u_i, \Delta_i, x_i\}_{i=1}^N$ and censoring is random $T \perp C|X$. Under these assumptions, and with $p(t|X)$ denotating the mass or probability density function (PDF) evaluated at t, the survival likelihood function with a parameter θ is proportional to (Kalbfleisch and Prentice, 2011):

$$\Pi_{i=1}^N p_\theta(u_i|x_i)^{\Delta_i} S_\theta(u_i|x_i)^{1-\Delta_i}.$$

In this work, we use the log-likelihood as a training objective function.

2.2. Related Work

Traditional Survival Analysis Traditionally, survival analysis makes distributional assumptions. The Cox model (Cox, 1972) makes the proportional hazard assumption. The accelerated failure time (AFT) model (Buckley and James, 1979; Wei, 1992) assumes that $\log(T) = X^T\theta + \epsilon$, where ϵ follows a log-logistic distribution. Multiple variants of Cox and AFT models (Aalen, 1980; Bennett, 1983; Cheng et al., 1995; Lin and Ying, 1995; Kalbfleisch and Prentice, 2011; Wu and Witten, 2019) have been proposed to introduce time-varying functions or different distributions. However, these extensions only use linear or simple non-linear models which may not be flexible enough to model complex data distributions. Avati et al. (2020) use deep networks to produce the parameters of a lognormal. Though this can capture nonlinear dependence of the lognormal's parameters on the input, the lognormal assumption may not be appropriate, e.g., if the true conditional distribution has more than one mode.

Deep Cox Models The Cox model has been extended with deep networks in several ways. DeepSurv (Katzman et al., 2018) uses a neural network to model the relative risk $g(X; \theta)$. Cox-Time (Kvamme et al., 2019) further allows the relative risk to depend on time

t. Kvamme and Borgan (2019) assume the hazard is constant in predefined time intervals. Nagpal et al. (2021) uses a mixture of Cox models parameterized by neural networks. These models optimize the partial likelihood function which does not require the access to survival functions. The partial likelihood is defined by

$$\Pi_{i:\Delta_i=1} \frac{\exp(g(u_i, x_i; \theta))}{\sum_{j \in R_i} \exp(g(u_i, x_j; \theta))},$$

where $R_i = \{j : y_j \geq y_i\}$, called the risk set, denotes the set of patients who survive at least as long as the i-th patient. The goal of maximizing the partial likelihood is to make patient i's relative risk at u_i greater than that of the other patients who survive longer. When there are thousands of datapoints, stochastic gradients are an efficient alternative to gradient computation for maximizing likelihoods. However, risk sets require the whole dataset to evaluate since the risk set involves all patients. This disadvantage precludes the use of stochastic gradient descent for training. Though we can use mini-batches of patients to approximate the risk set R_i, there are no theoretical guarantees for convergence.

Deep Discrete Models Deep categorical survival models (Miscouridou et al., 2018; Fotso, 2018; Goldstein et al., 2020) divide the time axis into a sequence of bins and turn survival analysis into predicting a time's bin. These models use K bins where the last bin includes all times greater than some value. DeepHit (Lee et al., 2018) adds a rank-based loss and uses discrete models for competing risks. Nnet-survival (Biganzoli et al., 1998; Gensheimer and Narasimhan, 2019) models the survival function by multiplications of conditional probabilities in previous time bins. These discrete models can approximate arbitrary smooth distributions with increasing fidelity as K increases (Miscouridou et al., 2018).

However, discrete models have their own problems. These models do not define what happens to the survival function estimation within a bin, at least without additional assumptions e.g. linearly interpolating the CDF. Next, it is challenging to choose the bin boundaries; it is unclear whether to set them by population percentiles or by regular intervals (Kvamme and Borgan, 2019; Tang et al., 2020; Craig et al., 2021). Using regular intervals may lead times to concentrate into a small subset of bins. For percentiles, it is unclear whether we should include the censored times into the population. Percentiles of the observed failure times may not equal the percentiles of true failure times. Finally, deep discrete models are based on classification architectures, meaning that they may be overconfident and suffer the same poor calibration observed for deep classifiers (Guo et al., 2017), as shown for survival analysis in Goldstein et al. (2020).

ODE-based Models Recently, continuous time models with neural ODEs have been proposed (Chen et al., 2018). SODEN (Tang et al., 2020) considers the evolution of cumulative hazard functions as an ODE while Danks and Yau (2022) model the CDF by an ODE. Groha et al. (2020) use ODEs for multi-state survival analysis. ODE-based models have tractable PDFs and CDFs. However, training neural ODEs is slow (Kelly et al., 2020) because of the expensive numerical integration inside ODE solvers. ODE-based models also involve extra hyperparameters related to ODE-solvers, including the solver type and tolerance level.

Other Deep Models Chapfuwa et al. (2018) use GANs for survival distribution modeling. But they do not use the likelihood as an objective for training since the PDF and CDF of GANs are intractable. The alternative, minimax training of GANs, is known to be unstable (Kodali et al., 2017; Bottou et al., 2018). Ranganath et al. (2016) use deep exponential families (Ranganath et al., 2015) with Weibull likelihoods. This approach necessitates the use of black-box variational inference with Monte Carlo gradients (Ranganath et al., 2014; Mohamed et al., 2020), which typically yields both a lower bound on the likelihood and noisier, slower optimizations. Survival stacking (Craig et al., 2021) casts the survival analysis as a classification task by predicting whether one patient is in other patients' risk sets. But for N datapoints, survival stacking creates $O(N^2)$ classification problems which is not tractable for large datasets.

Our Model In this work, we propose a new flexible survival model named Survival Mixture Density Networks. Survival MDNs utilize mixture density networks (Bishop, 1994) to allow flexible modeling. With Gaussians as the base distributions, computing the model CDF and PDF requires the evaluation of standard functions and the error function. The error function can be obtained efficiently via common approximations (Abramowitz et al., 1988) and Gaussian CDFs are implemented in most packages. Our simple approach can be trained through stochastic gradient descent and much faster than ODE-based models. We compare our model with previous approaches in table 1. In summary, we propose a continuous-time model that can be trained with stochastic gradients, without numerical ODE solving, and that moves beyond common modeling restrictions (e.g. that the density is log-normal or Cox).

Model	Flexible	Continuous-time	SGD	Without ODE-Solver
Cox	✗	✓	✗	✓
DeepSurv	✗	✓	✗	✓
DeepHit	✓	✗	✓	✓
Nnet-survival	✓	✗	✓	✓
Cox-Time	✓	✓	✗	✓
SODEN	✓	✓	✓	✗
Survival MDN	✓	✓	✓	✓

Table 1: Comparison of Different Models

3. Survival Mixture Density Networks

Our purpose is to build a survival model that has the following properties:

1. It has a differentiable PDF which can be evaluated efficiently.
2. It has a differentiable CDF which can be evaluated efficiently.
3. It is flexible enough to approximate a broad class of conditional time-to-event distributions $p(t|x)$ with support over $\mathbb{R}^+$.

The first two properties enable efficient training using maximum likelihood and using stochastic gradients. Examples of the last property are models that do not make assumptions like lognormality or proportional hazards, or that can capture multiple modes.

3.1. Mixture Density Networks

Mixture Density Networks (MDNs) (Bishop, 1994) form the key part of Survival MDNs. For a given x, MDNs model the conditional distribution $p(y|x)$ by mapping x through a neural network to produce the weights and parameters of a mixture model. Mixture density networks are flexible approximators; for any given x, with enough components, MDNs can approximate a broad class of conditional densities $p(y|x)$ as closely as desired (Bishop, 1994).

In this work, we use Gaussian mixtures (Reynolds et al., 2000; Reynolds, 2009). A discussion on different base distributions can be found in appendix C. Assume we have K components with weights $\{w_i\}_{i=1}^K$, means $\{\mu_i\}_{i=1}^K$ and standard deviations $\{\sigma_i\}_{i=1}^K$ such that $\sum_{i=1}^K w_i = 1$. The PDF of the Gaussian Mixture Density Network is given by

$$p\left(y|\{w_i, \mu_i, \sigma_i\}_{i=1}^K\right) = \sum_{i=1}^K w_i \mathcal{N}\left(y|\mu_i, \sigma_i^2\right),$$

where we denote $\mathcal{N}(y|\mu_i, \sigma_i^2)$ as the density of a Gaussian distributed random variable with mean μ_i and variance σ_i^2.

In mixture density networks, we build the conditional distribution by mapping the covariates x to parameters of the Gaussian Mixture Model through deep neural networks:

$$\{w_i(x), \mu_i(x), \sigma_i(x)\}_{i=1}^K = f_\theta(x),$$

where f_θ is a trainable neural network with parameters θ.

3.2. Survival Mixture Density Networks

We propose Survival Mixture Density Networks (Survival MDNs) to satisfy the properties we want for a survival model.

The sampling process for Survival MDN on a given input x is

1. Calculate $\{w_i(x), \mu_i(x), \sigma_i(x)\}_{i=1}^K = f_\theta(x)$.

2. Sample y according to the PDF $\sum_{i=1}^K w_i \mathcal{N}\left(y|\mu_i, \sigma_i^2\right)$. To do so, first sample a component i with probability equal to w_i and then sample from $\mathcal{N}(\mu_i, \sigma_i^2)$.

3. Map y to the event time t using $t = g(y) = \log(1 + \exp(y))$.

The invertible `softplus` function $g(y) = \log(1+\exp(y))$ maps the sample from the mixture density network to the positive domain. Another common choice to map the input from $\mathbb{R}$ to $\mathbb{R}^+$ is `exp`. We choose `softplus` over `exp` for the reason that `exp` may place high density on very large times.

Next, we show that the PDF and CDF of the Survival MDN is easy to compute. By the change of variables, the Survival MDN PDF at time t for input x is:

$$p(t|x) = \left|\frac{dg^{-1}(t)}{dt}\right|\Big(\sum_{i=1}^{K} w_i(x)\mathcal{N}\left(g^{-1}(t)|\mu_i(x), \sigma_i^2(x)\right)\Big).$$

For the simple choice of the `softplus`, the absolute value term does not depend on the parameters of neural network f_θ so this term does not contribute to gradients used for log-likelihood training. The Survival MDN CDF at time t can be computed easily as well. Denote $F(\cdot|\mu_i, \sigma_i^2)$ as the CDF of the i-th component in the Gaussian mixture model. Denote $F(t|x)$ as the CDF of the Survival MDN and $F_{\text{MDN}}(y|x)$ as the CDF of the underlying MDN. Since `softplus` is an increasing invertible function, we show that the CDF of the Survival MDN at time t only requires evaluations of the underlying Gaussian CDFs:

$$\begin{aligned}
F(t|x) &= F_{\text{MDN}}(g^{-1}(t)|x) \\
&= \int_{-\infty}^{g^{-1}(t)} \sum_{i=1}^{K} w_i(x)N\left(y|\mu_i(x), \sigma_i^2(x)\right) dy \\
&= \sum_{i=1}^{K} w_i(x) \int_{-\infty}^{g^{-1}(t)} N\left(y|\mu_i(x), \sigma_i^2(x)\right) dy \\
&= \sum_{i=1}^{K} w_i(x) F\left(g^{-1}(t)|\mu_i(x), \sigma_i^2(x)\right)
\end{aligned}$$

The evaluation of Gaussian CDFs can be done efficiently through the error function $\texttt{erf}(\cdot)$ which is the CDF of the standard normal distribution:

$$F\left(g^{-1}(t)|\mu_i(x), \sigma_i^2(x)\right) = \texttt{erf}\left(\left(g^{-1}(t) - \mu_i(x)\right)/\sigma_i(x)\right).$$

The `erf` function can be computed efficiently via common approximations (Abramowitz et al., 1988) and the Gaussian CDF is implemented in most packages. Now we have satisfied the first two desired properties (PDF and CDF). The last property, flexibility, follows since the Survival MDN maps time-to-event densities to densities over the reals via $y = g^{-1}(t)$ and a mixture density network with enough components and a wide and deep enough network can approximate a broad class of smooth densities $p(y|x)$ as closely as desired (Bishop, 1994). For tabular data, the network in MDN is a feedforward neural network. Other types of networks can also be used. For example, for image data, one can use convolutional neural networks and for text data one can use transformers. Instead of logits for classification, these models produce the parameters of the Gaussian Mixture at the last layer in MDNs.

4. Simulation Study

In this simulation experiment, we test Survival MDN and SODEN on a dataset where the proportional hazard assumption does not hold. We follow the simple simulation setting in

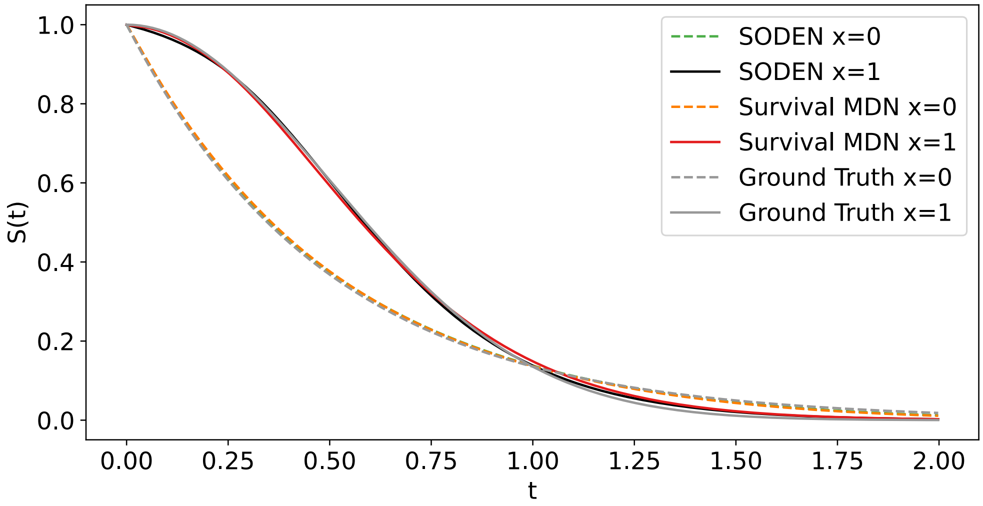

Figure 1: The survival functions of two groups. We show the survival function of two trained model SODEN and Survival MDN together with the ground truth on $x = 0$ and $x = 1$ separately. For $x = 0$, three survival functions are so close to each other that the green curve for SODEN is covered by the blue curve for Survival MDN and the gray curve for the ground truth.

SODEN (Tang et al., 2020). There are two group of x's, $x = 0$ and $x = 1$, and the ground truth survival function is:

$$S(t|x) = \exp(-2t) \cdot I\{x = 0\} + \exp(-2t^2) \cdot I\{x = 1\},$$

where I is the indicator function. The survival curves of the two groups cross so this survival distribution does not obey the proportional hazard (PH) assumption. Therefore, models that require the PH assumption cannot fit this dataset well. We generate x from a Bernoulli distribution with probability 0.5 and then generate t using the inverse CDF method. We sample the censored time uniformly on $[0, 2]$. Instead of simulating a fixed dataset, we use an "online" training method; in each iteration, we generate a new set of 1024 datapoints. We use the likelihood function for training. We train for 10,000 iterations for both SODEN and Survival MDN.

We show the resulting survival functions and ground truth in fig. 1. Both Survival MDN and SODEN's survival functions are close to the ground truth at both $x = 0$ and $x = 1$.

5. Real World Experiments

In this section, we compare Survival MDN with baselines Cox, DeepSurv, Cox-Time, Nnet-survival, DeepHit and SODEN. We use four different datasets: SUPPORT, METABRIC, GBSG and MIMIC. We evaluate all models on three different metrics: concordance, integrated binomial log-likelihood and integrated Brier score.

5.1. Datasets

We choose four different datasets: SUPPORT, METABRIC, GBSG and MIMIC. SUPPORT, METABRIC and GBSG are commonly used datasets for survival analysis, which can be found in the `pycox` package. MIMIC is a dataset we preprocessed from MIMIC-iv (Johnson et al., 2020) in PhysioNet (Goldberger et al., 2000). We describe the details of the datasets here:

- SUPPORT: the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment. It has 14 features. There are 8,873 datapoints, 32% of which are censored. We use the train/valid/test splits from SODEN (Tang et al., 2020)[2].
- METABRIC: the Molecular Taxonomy of Breast Cancer International Consortium. It has 9 features. There are 1,904 datapoints, 42% of which are censored. We use the train/valid/test splits from the SODEN repository.
- GBSG: The Rotterdam & German Breast Cancer Study Group. It has 7 features. There are 2,232 datapoints, 43% of which are censored.
- MIMIC: The Medical Information Mart for Intensive Care. The SODEN repository does not provide the data files for MIMIC. We choose patients that are alive 24 hours after admission to ICU. We define the event as mortality after admission. We define the censored time as the ICU discharged time. We collect time series features within the 24-hour window after the admission together with static features. For time series features, we use the minimum, mean and maximum within the window. We remove the features that are missing for more than half of the datapoints. Finally, we extract 65 features after preprocessing including common labs and vitals. There are 53,612 datapoints, 82% of which are censored. The SQL code that preprocesses the data from MIMIC-iv is attached in appendix A.

5.2. Baselines

We consider the following baseline models:

- Cox (Cox, 1972): A linear model with the proportional hazards assumption.
- DeepSurv (Katzman et al., 2018): A deep model with the linear function in Cox replaced by neural networks.
- Cox-Time (Katzman et al., 2018): A continuous time model that allows the relative risk in Cox to depend on time.
- Nnet-Survival (Gensheimer and Narasimhan, 2019): A discrete time model that models the conditional hazard in each time interval.
- DeepHit (Lee et al., 2018): A deep discrete time model that further adds a rank-based loss to the likelihood as the training objective.

[2] Available at https://github.com/jiaqima/SODEN

$P(C > \tau)$	Model	$C^{td}_{\tau}(\uparrow)$	$\text{IBLL}_{\tau}(\uparrow)$	$\text{IBS}_{\tau}(\downarrow)$
10^{-8}	Cox	0.596 ± .002	-0.568 ± .001	0.194 ± .001
	DeepSurv	0.609 ± .003	**-0.559** ± .002	**0.190** ± .001
	Cox-Time	0.607 ± .004	0.565 ± .002	0.191 ± .001
	Nnet-Survival	0.624 ± .003	-0.570 ± .004	0.193 ± .001
	DeepHit	**0.631** ± .003	-0.583 ± .006	0.197 ± .001
	SODEN	0.627 ± .003	-0.563 ± .002	0.191 ± .001
	Survival MDN	0.628 ± .003	**-0.559** ± .002	**0.190** ± .002
0.2	Cox	0.596 ± .002	-0.585 ± .001	0.201 ± .000
	DeepSurv	0.609 ± .003	-0.577 ± .002	0.197 ± .001
	Cox-Time	0.606 ± .004	-0.583 ± .002	0.199 ± .001
	Nnet-Survival	0.623 ± .003	-0.586 ± .003	0.201 ± .001
	DeepHit	**0.630** ± .003	-0.601 ± .006	0.205 ± .002
	SODEN	**0.630** ± .003	-0.601 ± .006	0.205 ± .002
	Survival MDN	0.628 ± .003	**-0.575**± .002	**0.196** ± .001
0.4	Cox	0.595 ± .002	-0.602 ± .001	0.208 ± .001
	DeepSurv	0.608 ± .002	-0.595 ± .002	0.205 ± .001
	Cox-Time	0.605 ± .004	-0.601 ± .002	0.207 ± .001
	Nnet-Survival	0.623 ± .003	-0.602 ± .003	0.208 ± .001
	DeepHit	**0.630** ± .003	-0.619 ± .007	0.212 ± .002
	SODEN	0.626 ± .003	-0.597 ± .002	0.205 ± .001
	Survival MDN	0.628 ± .003	**-0.593** ± .001	**0.204** ± .001

Table 2: Evaluation of all models on SUPPORT with concordance (C^{td}_{τ}), integrated binomial log-likelihood (IBLL_{τ}) and integrated Brier score (IBS_{τ}). The **bold** number indicates the best performance. We report mean ± standard error on all metrics.

- SODEN (Tang et al., 2020): An ODE-based continous time model.

For Cox, we use the implementation in the Python package `lifelines`. For DeepSurv, Cox-Time, Nnet-Survival and DeepHit, we use the implementations in the Python package `pycox`. For SODEN, we use the implementation from the SODEN repository.

5.3. Evaluation Metrics

We use the same evaluation metrics as SODEN (Tang et al., 2020). They are concordance, integrated binomial log-likelihood and Brier score. The implementations can be found in the SODEN repository. We briefly describe the three metrics here and refer to Tang et al. (2020) for more detailed descriptions.

Concordance The concordance index is originally proposed by Harrell Jr et al. (1984). It measures the probability that the relative order of the event time of two observations matches the predicted survival probabilities. Antolini et al. (2005) further relaxes the proportional hazard assumption in Harrell's concordance to create time dependent concordance.

$P(C > \tau)$	Model	$C^{td}_\tau(\uparrow)$	$\text{IBLL}_\tau(\uparrow)$	$\text{IBS}_\tau(\downarrow)$
10^{-8}	Cox	0.644 ± .006	-0.508 ± .009	0.169 ± .002
	DeepSurv	0.635 ± .007	-0.517 ± .011	0.171 ± .003
	Cox-Time	0.648 ± .007	-0.511 ± .009	0.172 ± .003
	Nnet-Survival	0.666 ± .005	-0.510 ± .007	0.171 ± .002
	DeepHit	**0.674** ± .006	-0.514 ± .004	0.174 ± .002
	SODEN	0.661 ± .005	-0.498 ± .008	0.167 ± .003
	Survival MDN	0.667 ± .004	**-0.489** ± .005	**0.165** ± .002
0.2	Cox	0.639 ± .006	-0.521 ± .006	0.176 ± .002
	DeepSurv	0.635 ± .006	-0.530 ± .005	0.179 ± .002
	Cox-Time	0.647 ± .005	-0.531 ± .007	0.179 ± .002
	Nnet-Survival	0.662 ± .004	-0.523 ± .003	0.177 ± .001
	DeepHit	**0.671** ± .004	-0.533 ± .003	0.182 ± .001
	SODEN	0.659 ± .003	-0.516 ± .006	0.174 ± .002
	Survival MDN	0.662 ± .004	**-0.510** ± .003	**0.172** ± .001
0.4	Cox	0.637± .006	-0.521 ± .006	0.175 ± .002
	DeepSurv	0.635 ± .006	-0.526 ± .005	0.178 ± .002
	Cox-Time	0.644 ± .005	-0.526 ± .006	0.178 ± .002
	Nnet-Survival	0.660 ± .003	-0.519 ± .003	0.176 ± .001
	DeepHit	**0.668** ± .003	-0.528 ± .003	0.180 ± .001
	SODEN	0.658 ± .004	-0.528 ± .003	0.180 ± .001
	Survival MDN	0.660 ± .002	**-0.508** ± .003	**0.172** ± .001

Table 3: Evaluation of all models on METABRIC with concordance (C^{td}_τ), integrated binomial log-likelihood (IBLL_τ) and integrated Brier score (IBS_τ). We report truncated metrics for τ's satisfying $P(C > \tau) = 10^{-8}, 0.2, 0.4$. The **bold** number indicates the best performance. We report mean ± standard error on all metrics.

Building off the inverse-weighting method in Cheng et al. (1995), Uno et al. (2011) introduces inverse probability weighted concordance to remove the dependence on the censoring distribution. They use the survival function of the censoring time $G(t) = P(C > t)$ as the weight and the Kaplan-Meier estimator for $G(t)$. Under the completely random censoring assumption $C \perp\!\!\!\perp (T, X)$, the inverse probability weighted estimator is consistent. This assumption is routinely made for evaluation, e.g. in Kvamme et al. (2019); Tang et al. (2020); Han et al. (2021). Due to the limited number of observations, the estimator of the inverse weight $1/\hat{G}(t)$ may be very large for some large-enough t. So Uno et al. (2011) introduce a truncated version of the concordance estimator within a pre-specified time interval $[0, \tau]$:

$$C^{td}_\tau = \frac{\sum_{i:\Delta_i=1, u_i<\tau} \sum_{j, u_i<u_j} I\left(\hat{S}(u_i|x_i) < \hat{S}(u_i|x_j)\right) / \hat{G}^2(u_i)}{\sum_{i:\Delta_i=1, u_i<\tau} \sum_{j:u_i<u_j} 1/\hat{G}^2(u_i)},$$

where $I(\cdot)$ is the indicator function. Here τ is used to truncate the large times that have very small $\hat{G}(t)$. We choose three τ's that satisfy $\hat{G}(\tau) = 10^{-8}, 0.2, 0.4$. When $\hat{G}(\tau) = 10^{-8}$, the truncated concordance is almost equal to the non-truncated version.

Integrated Brier Score The Brier score (BS) measures the mean square error between the ground-truth label and the predicted probability for a binary classifier. It measures both the calibration and discriminative performance (DeGroot and Fienberg, 1983). In survival analysis, we evaluate the Brier score at a given time t. The label is whether the patient survives after time t and the predicted probability is the survival function. We also consider an inverse probability weighted estimator (Graf et al., 1999; Gerds and Schumacher, 2006) for the Brier score at time t:

$$\mathrm{BS}(t) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{\hat{S}^2(t|x_i)I(u_i \le t, \Delta_i = 1)}{\hat{G}(u_i)} + \frac{(1-\hat{S}(t|u_i))^2 I(u_i > t)}{\hat{G}(t)}\right\}.$$

To consider all times, we use an integrated BS (IBS) over time interval $[0, \tau]$:

$$\mathrm{IBS}_\tau = \frac{1}{\tau}\int_0^\tau \mathrm{BS}(t)dt.$$

To avoid extreme inverse weights, we also report results for τ's that satisfy $\hat{G}(\tau) = 10^{-8}, 0.2, 0.4$. When $\hat{G}(\tau) = 10^{-8}$, τ is almost equal to the maximum time in the data.

Integrated Binomial Log-Likelihood Another common metric for survival analysis is the integrated binomial log-likelihood (IBLL). Different from IBS, IBLL uses binomial (Bernoulli) log-likelihood at each time step t:

$$\mathrm{BLL}(t) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{\log(1-\hat{S}(t|x_i)I(u_i \le t, \Delta_i = 1)}{\hat{G}(u_i)} + \frac{\log(\hat{S}(t|x_i)I(u_i > t)}{\hat{G}(t)}\right\}.$$

The IBLL is defined by:

$$\mathrm{IBLL}_\tau = \frac{1}{\tau}\int_0^\tau \mathrm{BLL}(t)dt.$$

We also report results for τ's satisfying $\hat{G}(\tau) = 10^{-8}, 0.2, 0.4$.

5.4. Experimental Setup

We randomly split datasets into training, validation, and testing sets. We use the validation set to choose the best epoch from training and hyperparameters and report the results on the test set. For SUPPORT/METABRIC/GBSG, we use 10 splits (8 for training, 1 for validation and 1 for test). For MIMIC, we use 5 splits (3 for training, 1 for validation, and 1 for test) since MIMIC is a larger dataset. We use random search to create 100 independent trials for different hyperparameters. We use the optimizer RMSProp (Tieleman et al., 2012).

For Survival MDN, following Sudarshan et al. (2020), we use a three-layer neural network that maps the features to a latent representation, and then from the latent representation we use three layers to output w's, μ's, σ's separately. We use a `softmax` layer to ensure that the sum of w's equals one and use an `exp` function to ensure the standard deviations σ's are

$P(C > \tau)$	Model	$C^{td}_{\tau}(\uparrow)$	$\text{IBLL}_{\tau}(\uparrow)$	$\text{IBS}_{\tau}(\downarrow)$
10^{-8}	Cox	0.645 ± .009	-0.523 ± .009	0.177 ± .004
	DeepSurv	0.663 ± .007	-0.509 ± .010	**0.172** ± .004
	Cox-Time	0.654 ± .007	-0.521 ± .009	0.176 ± .003
	Nnet-Survival	0.661 ± .006	-0.516 ± .008	0.174 ± .005
	DeepHit	0.665 ± .008	**-0.504** ± .017	0.176 ± .005
	SODEN	0.661 ± .012	-0.514 ± .017	0.173 ± .004
	Survival MDN	**0.668** ± .007	**-0.504** ± .006	**0.172** ± .003
0.2	Cox	0.645 ± .009	-0.519 ± .007	0.176 ± .003
	DeepSurv	0.663 ± .007	-0.505 ± .008	0.170 ± .002
	Cox-Time	0.654 ± .007	-0.517 ± .006	0.175 ± .002
	Nnet-Survival	0.661 ± .006	-0.509 ± .006	0.170 ± .003
	DeepHit	0.665 ± .008	-0.510 ± .008	0.172 ± .004
	SODEN	0.661 ± .012	-0.510 ± .009	0.172 ± .004
	Survival MDN	**0.668** ± .007	**-0.501** ± .006	**0.168** ± .002
0.4	Cox	0.645± .009	-0.519 ± .007	0.176 ± .003
	DeepSurv	0.663± .007	-0.505 ± .008	0.170 ± .002
	Cox-Time	0.654 ± .007	-0.517 ± .006	0.175 ± .002
	Nnet-Survival	0.661 ± .006	-0.509 ± .007	0.170 ± .003
	DeepHit	0.665 ± .008	-0.510 ± .008	0.172 ± .004
	SODEN	0.661 ± .012	-0.510 ± .009	0.172 ± .004
	Survival MDN	**0.668** ± .007	**-0.500** ± .006	**0.168** ± .002

Table 4: Evaluation of all models on GBSG with concordance (C^{td}_{τ}), integrated binomial log-likelihood (IBLL_{τ}) and integrated Brier score (IBS_{τ}). We report truncated metrics for τ's satisfying $P(C > \tau) = 10^{-8}, 0.2, 0.4$. The **bold** number indicates the best performance. We report mean ± standard error on all metrics.

positive. We vary the number of mixture components from 5 to 20. Different architectures can be used depending on the input type.

For other models, we vary the number of layers. Other hyperparameters include the hidden sizes, learning rate, batch normalization, momentum, dropout, and batch size. For DeepHit and Nnet-Survival, we vary the number of time intervals in addition. For other hyperparameters, we use the same tuning ranges as in Tang et al. (2020). We show the tuning ranges in appendix B.

5.5. Results

We report the results on the four datasets in table 2 (SUPPORT), table 3 (METABRIC), table 4 (GBSG), and table 5 (MIMIC). For SUPPORT and METABRIC, we use the exact same splits as the SODEN repository so we use their results for the baselines.

For concordance, DeepHit has the best concordance on SUPPORT and METABRIC while the continuous time model Survival MDN has the best concordances on GBSG and

MIMIC. For IBLL and IBS, Survival MDN has the best performance across all datasets. The IBLL and IBS care more about the exact survival probability prediction at each time. The discrete time model DeepHit may not yield an accurate estimate of the survival probability for a particular time since it does not distinguish the times inside one time interval. For the discrete time models, it is also challenging to choose the bin boundaries (Kvamme and Borgan, 2019; Tang et al., 2020; Craig et al., 2021). The discrete models' concordance on MIMIC is worse than that of SODEN and Survival MDN. Continuous time models Survival MDN and SODEN have similar performance on concordance on the four datasets since they are both flexible continuous time models. There is little difference among $\hat{G}(\tau) = 10^{-8}, 0.2, 0.4$ for the concordance, IBLL and IBS on small datasets SUPPORT, METABRIC and GBSG, which is the same observation in SODEN (Tang et al., 2020).

$P(C > \tau)$	Model	$C_\tau^{td}(\uparrow)$	$\text{IBLL}_\tau(\uparrow)$	$\text{IBS}_\tau(\downarrow)$
10^{-8}	Cox	0.642 ± .002	-0.211 ± .001	0.061 ± .001
	DeepSurv	**0.663** ± .001	-0.212 ± .003	0.061 ± .001
	Cox-Time	0.653 ± .001	-0.210 ± .003	0.061 ± .001
	Nnet-Survival	0.649 ± .002	-0.206 ± .000	0.061 ± .001
	DeepHit	0.647 ± .002	-0.206 ± .001	0.061 ± .001
	SODEN	0.659 ± .002	**-0.204** ± .002	0.060 ± .001
	Survival MDN	0.660 ± .002	**-0.204** ± .002	**0.059** ± .001
0.2	Cox	0.711 ± .004	-0.473 ± .133	0.091 ± .014
	DeepSurv	0.734 ± .003	-0.462 ± .150	0.089 ± .015
	Cox-Time	0.726 ± .002	-0.443 ± .126	0.061 ± .001
	Nnet-Survival	0.722 ± .004	-0.229 ± .004	0.066 ± .001
	DeepHit	0.719 ± .004	-0.233 ± .004	0.066 ± .001
	SODEN	0.733 ± .002	-0.229 ± .004	**0.065** ± .001
	Survival MDN	**0.736** ± .003	**-0.228** ± .004	**0.065** ± .001
0.4	Cox	0.780 ± .002	-0.588 ± .136	0.071 ± .031
	DeepSurv	0.797 ± .001	-0.423 ± .202	0.045 ± .018
	Cox-Time	0.790 ± .002	-0.501 ± .267	0.037 ± .010
	Nnet-Survival	0.784 ± .003	-0.082 ± .003	**0.018** ± .001
	DeepHit	0.787 ± .003	-0.083 ± .002	0.019 ± .001
	SODEN	**0.805** ± .005	-0.084 ± .002	0.019 ± .001
	Survival MDN	**0.805** ± .001	**-0.078** ± .002	**0.018** ± .001

Table 5: Evaluation of all models on MIMIC with concordance (C_τ^{td}), integrated binomial log-likelihood (IBLL_τ) and integrated Brier score (IBS_τ). We report truncated metrics for τ's satisfying $P(C > \tau) = 10^{-8}, 0.2, 0.4$. The **bold** number indicates the best performance. We report mean ± standard error on all metrics.

The training time of SODEN is much longer than Survival MDN. We collect the training time of two models with the same hidden size 32 and number of layers 4 on METABRIC. We use the maximum number of components in the tuning range 20 for Survival MDN.

We show the test concordance versus the training time for Survival MDN and SODEN on `GeForce RTX 2080 Ti` in fig. 2. We can see that Survival MDN reached the peak of the test concordance much faster than SODEN. On average, each epoch of Survival MDN costs 0.20 seconds while each epoch of SODEN costs 23.82 seconds. Training Survival MDN is more than 100 time faster than SODEN.

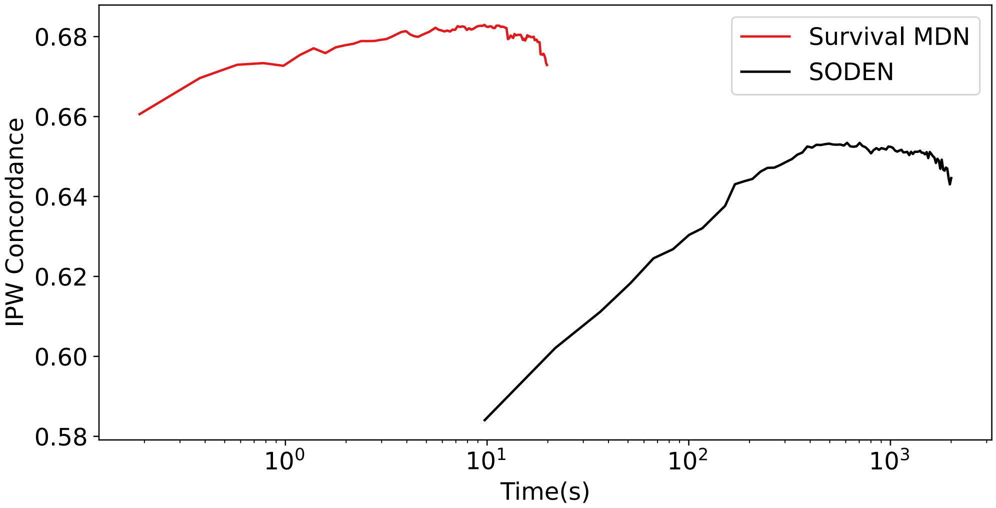

Figure 2: Comparison between Survival MDN and SODEN on IPW Concordance versus training time. The time is shown in log scale.

6. Discussion

Survival modeling plays an important role in risk estimation and clinical decision making. We propose Survival MDN, a simple flexible continuous time survival model. We combine two simple yet elegant tools—mixture densities and change of variables—to produce flexible survival models. While recent approaches achieve similar flexibility, it is achieved at the expense of training time, complexity, and inconvenient hyper-parameters. Without introducing such complexity, Survival MDNs achieve similar or better performance.

Limitations Currently, the proposed model, Survival MDN, mainly considers Gaussian Mixtures. Though Gaussian Mixtures have universal approximation power, a combination of different base distributions, e.g. generalized logistics, in mixture density networks may improve performance. Regarding experimental evaluation, the marginal censoring assumption used in the reweighting estimators is common practice in the literature, but may not be appropriate. Evaluation with censored data is impossible without assumptions, but it could be possible to improve evaluation by making conditional censoring assumptions.

Acknowledgments

This work was made possible by the following grants/awards:

- NIH/NHLBI Award R01HL148248
- NSF Award 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science.
- NSF CAREER Award 2145542

The authors thank Weijing Tang, Jiaqi Ma, Qiaozhu Mei and Ji Zhu for providing a great codebase. The authors thank Weijing Tang for a detailed explanation of the codebase.

References

Odd Aalen. A model for nonparametric regression analysis of counting processes. In *Mathematical statistics and probability theory*, pages 1–25. Springer, 1980.

Milton Abramowitz, Irene A Stegun, and Robert H Romer. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1988.

Tanvir Ahmad, Assia Munir, Sajjad Haider Bhatti, Muhammad Aftab, and Muhammad Ali Raza. Survival analysis of heart failure patients: A case study. *PloS one*, 12(7):e0181001, 2017.

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

Anand Avati, Tony Duan, Sharon Zhou, Kenneth Jung, Nigam H Shah, and Andrew Y Ng. Countdown regression: sharp and calibrated survival predictions. In *Uncertainty in Artificial Intelligence*, pages 145–155. PMLR, 2020.

Steve Bennett. Analysis of survival data by the proportional odds model. *Statistics in medicine*, 2(2):273–277, 1983.

Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine*, 17(10):1169–1186, 1998.

Christopher M Bishop. Mixture density networks. 1994.

Leon Bottou, Martin Arjovsky, David Lopez-Paz, and Maxime Oquab. Geometrical insights for implicit generative modeling. In *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*, pages 229–268. Springer, 2018.

Jonathan Buckley and Ian James. Linear regression with censored data. *Biometrika*, 66(3): 429–436, 1979.

Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Courtney Page, Benjamin Goldstein, Lawrence Carin Duke, and Ricardo Henao. Adversarial time-to-event modeling. In *International Conference on Machine Learning*, pages 735–744. PMLR, 2018.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

SC Cheng, LJ Wei, and Z Ying. Analysis of transformation models with censored data. *Biometrika*, 82(4):835–845, 1995.

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

Erin Craig, Chenyang Zhong, and Robert Tibshirani. Survival stacking: casting survival analysis as a classification problem. *arXiv preprint arXiv:2107.13480*, 2021.

Dominic Danks and Christopher Yau. Derivative-based neural modelling of cumulative distribution functions for survival analysis. 2022.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Javad Faradmal, Atefeh Talebi, Abbas Rezaianzadeh, and Hossein Mahjub. Survival analysis of breast cancer patients using cox and frailty models. *Journal of Research in Health Sciences*, 12(2):127–130, 2012.

Stephane Fotso. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprint arXiv:1801.05512*, 2018.

Michael F Gensheimer and Balasubramanian Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019.

Thomas A Gerds and Martin Schumacher. Consistent estimation of the expected brier score in general survival models with right-censored event times. *Biometrical Journal*, 48(6): 1029–1040, 2006.

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, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Mark Goldstein, Xintian Han, Aahlad Puli, Adler Perotte, and Rajesh Ranganath. X-cal: Explicit calibration for survival analysis. *Advances in neural information processing systems*, 33:18296–18307, 2020.

Peter Goldstraw, Kari Chansky, John Crowley, Ramon Rami-Porta, Hisao Asamura, Wilfried EE Eberhardt, Andrew G Nicholson, Patti Groome, Alan Mitchell, Vanessa Bolejack, et al. The iaslc lung cancer staging project: proposals for revision of the tnm stage groupings in the forthcoming (eighth) edition of the tnm classification for lung cancer. *Journal of Thoracic Oncology*, 11(1):39–51, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Stefan Groha, Sebastian M Schmon, and Alexander Gusev. A general framework for survival analysis and multi-state modelling. *arXiv preprint arXiv:2006.04893*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Xintian Han, Mark Goldstein, Aahlad Puli, Thomas Wies, Adler Perotte, and Rajesh Ranganath. Inverse-weighted survival games. *Advances in Neural Information Processing Systems*, 34, 2021.

Frank E Harrell Jr, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984.

Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A targeted real-time early warning score (trewscore) for septic shock. *Science translational medicine*, 7 (299):299ra122–299ra122, 2015.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and R Mark IV. Mimic-iv (version 0.4). *PhysioNet*, 2020.

Nicholas R Jones, Andrea K Roalfe, Ibiye Adoki, FD Richard Hobbs, and Clare J Taylor. Survival of patients with chronic heart failure in the community: a systematic review and meta-analysis. *European journal of heart failure*, 21(11):1306–1325, 2019.

John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*. John Wiley & Sons, 2011.

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.

Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33:4370–4380, 2020.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Fatemeh Kojoria, Rui Chen, Christopher A Caldarone, Sandra L Merklinger, Anthony Azakie, William G Williams, Glen S Van Arsdell, John Coles, and Brian W McCrindle. Outcomes of mitral valve replacement in children: a competing-risks analysis. *The Journal of Thoracic and Cardiovascular Surgery*, 128(5):703–709, 2004.

Håvard Kvamme and Ørnulf Borgan. Continuous and discrete-time survival prediction with neural networks. *arXiv preprint arXiv:1910.06724*, 2019.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *arXiv preprint arXiv:1907.00825*, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

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.

Dongyeol Lee, Hee Yeoun Kim, Jin Ho Lee, Yong Hun Sin, Joong Kyung Kim, and Joon Seok Oh. Long-term survival analysis of kidney transplant recipients receiving mizoribine as a maintenance immunosuppressant: a single-center study. In *Transplantation Proceedings*, volume 51, pages 2637–2642. Elsevier, 2019.

DY Lin and Zhiliang Ying. Semiparametric analysis of general additive-multiplicative hazard models for counting processes. *The annals of Statistics*, pages 1712–1734, 1995.

Shi-Woei Lin and Kartika Nur Anisa. Effects of socioeconomic status on cancer patient survival: Counterfactual event-based mediation analysis. *Cancer Causes & Control*, 32 (1):83–93, 2021.

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.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.

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, 2021.

David A Patterson and Myung-Shin Lee. Intensive case management and rehospitalization: a survival analysis. *Research on Social Work Practice*, 8(2):152–171, 1998.

Stuart J Pocock, Sheila M Gore, and Gillian R Kerr. Long term survival analysis: the curability of breast cancer. *Statistics in medicine*, 1(2):93–104, 1982.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep exponential families. In *Artificial Intelligence and Statistics*, pages 762–771. PMLR, 2015.

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

Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

S Rodrigues, R Escoli, C Eusébio, L Dias, M Almeida, LS Martins, S Pedroso, AC Henriques, and A Cabrita. A survival analysis of living donor kidney transplant. In *Transplantation Proceedings*, volume 51, pages 1575–1578. Elsevier, 2019.

Mukund Sudarshan, Wesley Tansey, and Rajesh Ranganath. Deep direct likelihood knockoffs. *Advances in neural information processing systems*, 33:5036–5046, 2020.

Weijing Tang, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu. Soden: A scalable continuous-time survival model through ordinary differential equation networks. *arXiv preprint arXiv:2008.08637*, 2020.

Weijing Tang, Kevin He, Gongjun Xu, and Ji Zhu. Survival analysis via ordinary differential equations. *Journal of the American Statistical Association*, (just-accepted):1–41, 2022.

Tijmen Tieleman, Geoffrey Hinton, et al. 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.

Jianyong Wang, Nan Chen, Jixiang Guo, Xiuyuan Xu, Lunxu Liu, and Zhang Yi. Survnet: A novel deep neural network for lung cancer survival analysis with missing values. *Frontiers in Oncology*, 10:588990, 2021.

Ru Wang, Yayun Zhu, Xiaoxu Liu, Xiaoqin Liao, Jianjun He, and Ligang Niu. The clinicopathological features and survival outcomes of patients with different metastatic sites in stage iv breast cancer. *BMC cancer*, 19(1):1–12, 2019.

Zhuo Wang, John S Ji, Yang Liu, Runyou Liu, Yuxin Zha, Xiaoyu Chang, Lun Zhang, Qian Liu, Yu Zhang, Jing Zeng, et al. Survival analysis of hospital length of stay of novel coronavirus (covid-19) pneumonia patients in sichuan, china. *Medrxiv*, 2020.

Lee-Jen 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.

WHO et al. *Prevention of cardiovascular disease: guidelines for assessment and management of total cardiovascular risk.* World Health Organization, 2007.

Peter WF Wilson, Ralph B D'Agostino, Daniel Levy, Albert M Belanger, Halit Silbershatz, and William B Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

Jiacheng Wu and Daniela Witten. Flexible and interpretable models for survival data. *Journal of Computational and Graphical Statistics*, 28(4):954–966, 2019.

Ting Yin, Shi Shi, Xu Zhu, Iokfai Cheang, Xinyi Lu, Rongrong Gao, Haifeng Zhang, Wenming Yao, Yanli Zhou, and Xinli Li. A survival prediction for acute heart failure patients via web-based dynamic nomogram with internal validation: A prospective cohort study. *Journal of Inflammation Research*, 15:1953, 2022.

Appendix A. MIMIC SQL code

```
select
-- ids
pat.subject_id as subject_id, adm.hadm_id as hadm_id,icu.stay_id as stay_id,
-- demographics
CASE WHEN pat.gender="M" THEN 1 ELSE 0 END as is_male,
CASE WHEN adm.ethnicity="WHITE" THEN 1 ELSE 0 END as is_white,
icu_detail.admission_age as age,
-- weight height
fdw.weight ,
fdh.height ,
-- LOS
icu.los as los_icu_days,
icu_detail.los_hospital as los_hosp_days,
-- death
--icu_detail.icu_intime as icu_intime,
--icu_detail.dod as dod,
TIMESTAMP_DIFF(icu_detail.dod, icu_detail.icu_intime, HOUR) / 24 as time_to_death,
case
    when icu_detail.dod is null then 0
    else 1
end
as death,
-- vitals labs min max mean
vitals.*,
labs.*,
sofa.*
from ‘physionet-data.mimic_core.patients‘ pat
inner join
    ‘physionet-data.mimic_core.admissions‘ adm
        on pat.subject_id=adm.subject_id
inner join
    ‘physionet-data.mimic_icu.icustays‘ icu
      on adm.subject_id=icu.subject_id
      and
      adm.hadm_id=icu.hadm_id
```

```
inner join
     ‘physionet-data.mimic_derived.first_day_height‘ fdh
        on
        adm.subject_id = fdh.subject_id  and icu.stay_id = fdh.stay_id
inner join
     ‘physionet-data.mimic_derived.first_day_weight‘ fdw
        on
        adm.subject_id = fdw.subject_id  and icu.stay_id = fdw.stay_id
inner join
     ‘physionet-data.mimic_derived.icustay_detail‘ icu_detail
        on
        adm.subject_id=icu_detail.subject_id
        and
        adm.hadm_id=icu_detail.hadm_id
        and
        icu.stay_id=icu_detail.stay_id
inner join
    ‘physionet-data.mimic_derived.first_day_sofa‘ sofa
    on
        adm.subject_id=sofa.subject_id
        and
        adm.hadm_id=sofa.hadm_id
        and
        icu.stay_id=sofa.stay_id

inner join
     ‘physionet-data.mimic_derived.first_day_vitalsign‘ vitals
     on
    adm.subject_id=vitals.subject_id
    and
    icu.stay_id=vitals.stay_id
inner join
 ‘physionet-data.mimic_derived.first_day_lab‘ labs
     on
    adm.subject_id=labs.subject_id
    and
    icu.stay_id=labs.stay_id
where icu_detail.los_icu > 1
    and pat.gender is not null
    and adm.ethnicity is not null
    and adm.ethnicity != "UNABLE TO OBTAIN"
    and adm.ethnicity != "UNKNOWN"
```

Appendix B. Tuning Ranges of Hyperparameters

We show the search range of hyperparameters in table 6.

Batch size	$\{32, 64, 128, 256\}$ for METABRIC, GBSG
	$\{128, 256, 512\}$ for SUPPORT
	$\{512, 1024\}$ for MIMIC
Number of layers	$\{1, 2, 4\}$
Hidden size	$[2^2, 2^7]$
Learning rate	$[10^{-4.5}, 10^{-1.5}]$
Weight decay	$[10^{-9}, 10^{-4}]$
Momentum	$[0.85, 0.99]$
Dropout	$\{0, 0.1, 0.5\}$
Batch normalization	{True, False}
α (Surrogate ranking loss in DeepHit)	$[0, 1]$
σ (Surrogate ranking loss in DeepHit)	$\{0.25, 1, 5\}$
Number of intervals	{10, 50, 100, 200, 400} for SUPPORT, METABRIC, GBSG
(DeepHit, Nnet-survival)	{50, 100, 200, 400, 800} for MIMIC

Table 6: Tuning ranges of hyperparameters

Appendix C. Discussion on Different Base Distributions

Here we compare Gaussian base with an altenative base, the generalized logistic distribution, on marginal data generations. We use the following form of the generalized logistic distribution:

$$F(x; \alpha) = 1 - \frac{e^{-\alpha x}}{(1 + e^{-x})^{\alpha}}.$$

We also shift the generalized logistic distribution using scale and location. In this generalized logistic distribution, we have one more parameter α which can control the magnitude of the power.

We consider three different marginal data generation cases:

- LogNormal distribution with $\mu = 0.1$ and $\sigma = 0.1$. LogNormal distribution is a common one researchers use in survival analysis. The variance is small in this data generation distribution.
- Student T distribution with degree of freedom one and transformed to positive values through softplus. Student T distribution has a heavy tail.
- Gamma distribution with shape 0.1 and scale 1. When shape is smaller than one, the Gamma distribution put a lot of mass on values close to zero. This may be hard for a mixture model to fit.

We sample the censored time uniformly from $[0, 10]$. We still use an online training which generates a whole new batch data in every update step.

The results of fitting LogNormal data is shown in fig. 3. The Gaussian base has survival functions overlapping with the ground truth but the generalized logistic base cannot fit it well.

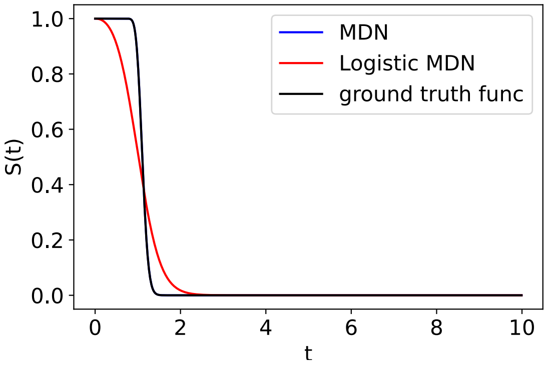

Figure 3: LogNormal Data.

The result of fitting student T data is shown in fig. 4. For heavy tailed student T, both Gaussian base and generalized logistic base can also fit it well with survival functions overlapping the ground truth.

The result of fitting Gamma data is shown in fig. 5. The generalized logistic base can fit the Gamma data well while there is some gap between the ground truth and the Gaussian

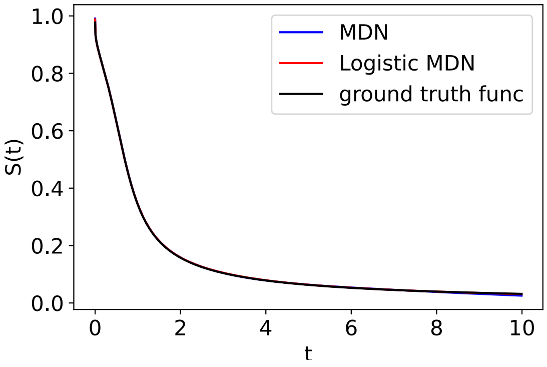

Figure 4: Student T + Softplus Data.

base survival function. In Gamma data with a small shape, the generalized logistic base is a better choice.

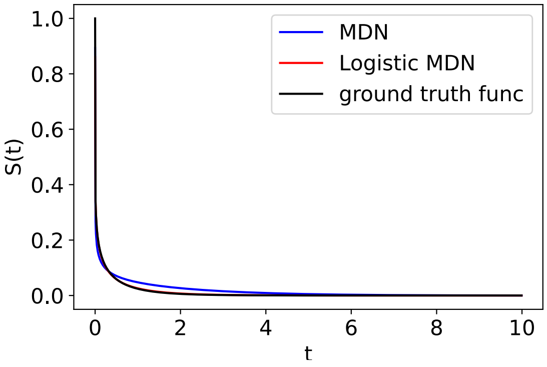

Figure 5: Gamma Data.