Neural Topic Models with Survival Supervision: Jointly Predicting Time-to-Event Outcomes and Learning How Clinical Features Relate

George H. Chen*[1], Linhong Li*[2], Ren Zuo[3], Amanda Coston[1], and Jeremy C. Weiss[1]

[1]Heinz College of Information Systems and Public Policy, Carnegie Mellon University
[2]McKinsey & Company
[3]Cornerstone Research

Abstract

We present a neural network framework for learning a survival model to predict a time-to-event outcome while simultaneously learning a topic model that reveals feature relationships. In particular, we model each subject as a distribution over "topics", where a topic could, for instance, correspond to an age group, a disorder, or a disease. The presence of a topic in a subject means that specific clinical features are more likely to appear for the subject. Topics encode information about related features and are learned in a supervised manner to predict a time-to-event outcome. Our framework supports combining many different topic and survival models; training the resulting joint survival-topic model readily scales to large datasets using standard neural net optimizers with minibatch gradient descent. For example, a special case is to combine LDA with a Cox model, in which case a subject's distribution over topics serves as the input feature vector to the Cox model. We explain how to address practical implementation issues that arise when applying these neural survival-supervised topic models to clinical data, including how to visualize results to assist clinical interpretation. We study the effectiveness of our proposed framework on seven clinical datasets on predicting time until death as well as hospital ICU length of stay, where we find that neural survival-supervised topic models achieve competitive accuracy with existing approaches while yielding interpretable clinical topics that explain feature relationships.

*equal contribution

1 Introduction

Predicting the amount of time until a critical event occurs—such as death, disease relapse, or hospital discharge—is a central focus in the field of survival analysis. Especially with the increasing availability of electronic health records, survival analysis data in healthcare often have both a large number of subjects and a large number of features measured per subject. In coming up with an interpretable survival analysis model to predict time-to-event outcomes for these large-scale datasets, a standard approach is to use the classical Cox proportional hazards model [Cox, 1972], possibly with features selected using lasso regularization [Simon et al., 2011] or stepwise regression [Harrell et al., 1984]. However, these Cox-based models do not inherently learn how features relate. Instead, to try to understand feature interactions with a Cox model, one would have to, for example, introduce a large number of features that encode interactions between the original features. This approach is impractical when the number of features is very large.

To simultaneously address the two objectives of learning a survival model for time-to-event prediction and learning how features relate through a topic model, Dawson and Kendziorski [2012] combine latent Dirichlet allocation (LDA) [Blei et al., 2003] with Cox proportional hazards to obtain a method they call SURVLDA. The idea is to represent each subject as a distribution over topics, and each topic as a distribution over which clinical feature values appear. For example, a topic could correspond to a severe disease state or a particular age group. The Cox model is given the subjects' distributions over topics as input rather than the subjects' raw feature vectors. Importantly, the topic and survival models are jointly learned.

In this paper, we propose a general framework for deriving neural survival-supervised topic models that is substantially more flexible than SURVLDA. Specifically, SURVLDA estimates model parameters via variational inference update equations derived specifically for LDA combined with the standard Cox model; to use another other sort of combination would require re-deriving the inference algorithm. Moreover, the inference algorithm for SURVLDA as stated in their paper does not easily scale to large datasets. In contrast, our approach combines essentially any topic model and any survival model that can be cast in a neural net framework (precise prerequisites of our framework are given in Section 2); combining LDA with the Cox proportional hazards model is only one special case. As a byproduct of taking a neural net approach, we can readily leverage many deep learning advances. For example, we can avoid deriving a special inference algorithm and instead use any neural net optimizer such as Adam [Kingma and Ba, 2014] to learn the joint model in minibatches, which readily scales to large datasets. Importantly, our framework yields survival-supervised topic models that are amenable to interpretation so long as the underlying topic and survival models are.

As numerous combinations of neural topic/survival models are possible, we only demonstrate four combinations, corresponding to combining either LDA or SAGE [Eisenstein et al., 2011] topic models with either the Cox pro-

portional hazards model or an accelerated failure time model (e.g., Cox 1972, Prentice 1978). We make these combinations within the SCHOLAR neural topic modeling framework by Card et al. [2018] and thus refer to the resulting neural survival-supervised topic models as SCHOLAR LDA-COX, SCHOLAR LDA-AFT, SCHOLAR SAGE-COX, and SCHOLAR SAGE-AFT; note that SCHOLAR LDA-COX is a neural network variant of SURVLDA. We benchmark the four neural survival-supervised models on seven datasets, finding that they can yield accuracy competitive with deep learning baselines [Katzman et al., 2018, Lee et al., 2018] while yielding interpretable topics. In contrast, the deep learning baselines are not interpretable.

Importantly, we discuss practical challenges encountered in learning these neural survival-supervised topic models on clinical data to obtain interpretable topics. For example, we found the standard approach in topic modeling of just listing the top features per topic to often not be interpretable because this listing does not explain how these top features' probabilities of appearing vary across topics. As an alternative, we propose a new heatmap visualization of learned topics that we found can better assist clinical interpretation. Separately, we find encouraging sparsity in learned topics to make the topics *less* interpretable. Our observation is that sometimes multiple clinical events/measurements are taken that altogether help explain a condition, whereas encouraging sparsity tends to only pick out one among multiple related features. This is essentially the same problem encountered when using lasso for linear regression: when there is a group of variables with high pairwise correlation, lasso arbitrarily chooses one of these variables [Zou and Hastie, 2005]. We do not want this sort of behavior when our goal is to understand how different features relate.

As a separate issue on interpretability, especially when the number of features is large, it is possible that many features do not help explain survival outcomes. Dawson and Kendziorski [2012] address this issue by using a preprocessing procedure for SURVLDA. Specifically, they cluster on the subjects' data based on their survival outcomes. Then they remove features that are not sufficiently different across the clusters. The issue with this approach is that it is ad hoc and how it impacts downstream analyses is unclear. Moreover, there are many possible clustering approaches that can be used each with its own (hyper)parameter settings that can be tuned. We do not use such a heuristic preprocessing step to filter features. Instead, we filter features *after* learning a survival-supervised topic model. This strategy has been demonstrated to work as well as filtering features *before* learning topic models [Schofield et al., 2017] although it has not been demonstrated in the survival analysis context. Filtering after learning the model is appealing since we can apply different filters (potentially with clinician input) without having to retrain the model. For example, we can screen out features that appear in too few or too many patients on demand after learning the model.

As a concrete example, on a cancer dataset where we aim to predict time until death, the topics learned by one of our neural survival-supervised topic models SCHOLAR LDA-COX are shown as a heatmap in Figure 1. In the heatmap, the columns correspond to different topics (ordered from left to right corre-

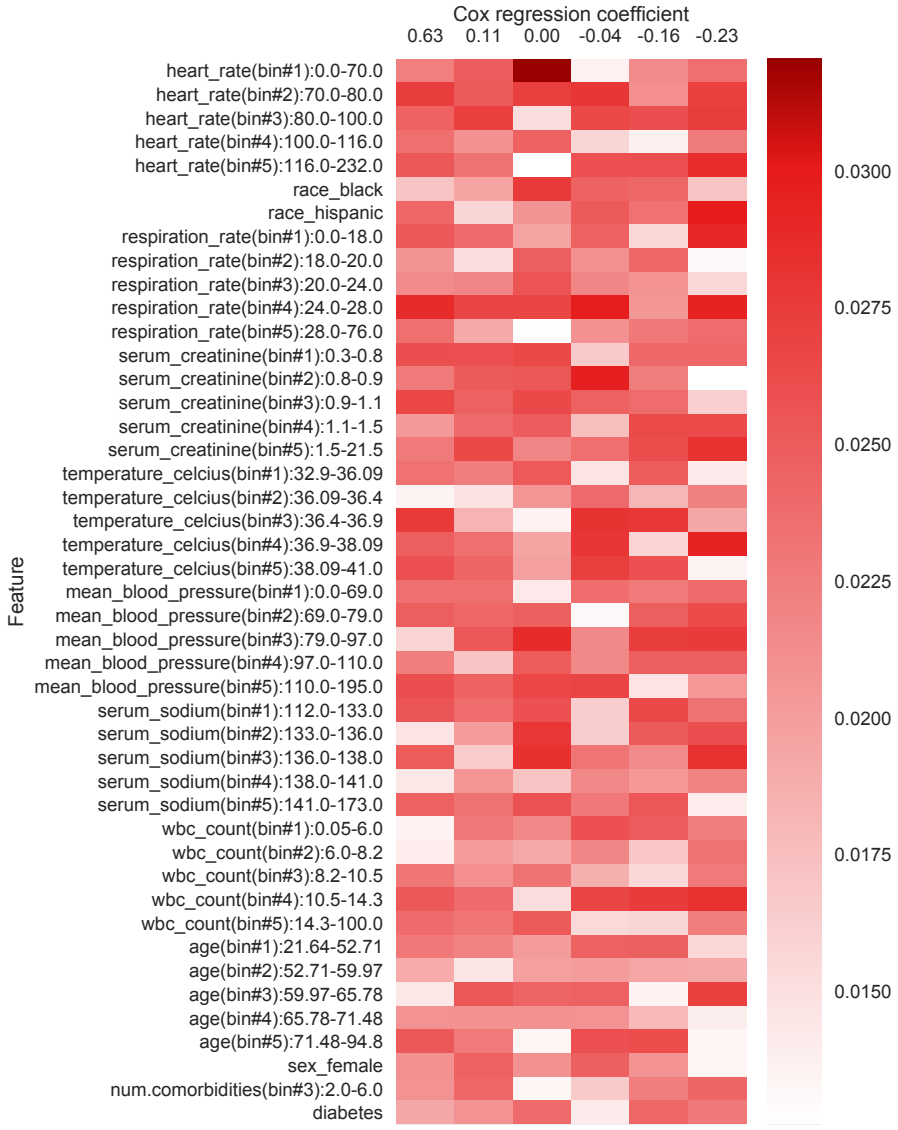


Figure 1: Topics learned by SCHOLAR LDA-COX on the SUPPORT3 (cancer) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic. Note that two different features that are highly probable (darker shade of red) for the same topic does *not* mean that they must co-occur when that topic is present, and it is possible that neither occurs. A helpful way to think about this is to consider how topic modeling works when applied to text data such as news articles. In this case, a learned topic might correspond to *sports*, which could have highly probable words such as "basketball" and "skiing". A text document could be about sports yet mentions neither of these words. This same idea applies to our setting where we represent patients in terms of clinical topics.

sponding to being associated with shorter to longer average survival time), the rows correspond to different clinical measurements (continuous measurements are discretized into bins), and the color values are probabilities where a deeper red roughly means that the feature is more prominent for a particular topic. We explain in Section 4 precisely how this heatmap is constructed and how the rows are ordered. By looking at this heatmap, we can quickly identify how feature occurrences tend to differ across the topics. We can interpret the topics by looking at which features tend to be highly probable for each topic. Our resulting interpretations are shown in Table 1.

Extremely importantly, the interpretation of the learned topics requires an abundance of caution. While our learned topic models are competitive with various state-of-the-art baselines in terms of prediction accuracy, the best accuracy scores possible are not high for the various prediction tasks we consider in our experiments. Thus, we cannot claim that the learned topics are "correct", and we believe that they require more extensive validation if they are to be deployed for clinical use. However, the learned topics can be very helpful in model debugging. By visualizing them with our heatmap strategy, we can spot inconsistencies between topics learned and clinical intuition, which could suggest ways to improve the model (e.g., adding additional constraints or regularization, changing specific data preprocessing steps). In contrast, state-of-the-art deep learning baselines that we benchmark against are not interpretable and do not provide straightforward visualizations to assist model debugging and improvement.

With the above disclaimer, if we suppose for the moment that the learned topics in Figure 1/Table 1 capture valid associations, then the topics could provide actionable insights. In the problem of predicting time until death for cancer patients, we may want to tease apart elderly cancer patients in terms of their risk of mortality. Topics 1, 4, and 5 (as numbered in Table 1) would be particularly relevant in this case as they focus more on elderly patients and are associated with different risks of mortality. By looking at what differentiates these topics, we see that fever, infection, and inflammation are key indicators, which we could consider interventions for. Note that whether a patient is more associated with topic 1 vs 5 can be distinguished by other characteristics such as blood pressure and white blood cell count. One might want to consider more aggressive interventions for patients mostly associated with topic 1 since their prognosis is worse collectively.

In summary, our main contributions are as follows:

- We propose a general neural network framework for combining neural topic models with survival models. This framework is meant for large datasets in which both the number of subjects and the number of features are large, where a key goal is to discover possible feature relationships.
- We discuss practical issues that arise when applying our framework to clinical data, including visualization strategies to assist clinical interpretation.

Table 1: Summary of topics learned by SCHOLAR LDA-COX on the SUPPORT3 (cancer) dataset. Higher Cox regression β coefficient is associated with shorter survival time.

Topic number	β	Topic interpretation
1	0.63	old otherwise normal
2	0.11	cardiorenal problems with comorbidities
3	0	baseline
4	-0.04	old, feverish, infection/inflammation
5	-0.16	old with inflammation
6	-0.23	normal healthier

- We experimentally show that neural survival-supervised topic models often work as well as deep learning baselines but have the added advantage of producing clinically interpretable topics. The deep learning baselines are not interpretable.

Outline The rest of the paper is organized as follows. We provide background and prerequisites of our framework in Section 2. We then explain how to construct neural-survival supervised topic models with an explicit background topic in Section 3, with examples given for how to combine LDA and SAGE topic models with the Cox and log-logistic accelerated failure time survival models. We then benchmark these models against classical and deep learning baselines in Section 4, where we also discuss model interpretability. We end the paper with a discussion in Section 5.

2 Background and Prerequisites for Our Framework

We begin with some background and notation, first stating the format of the data we assume we have access to. Then we review key ideas of topic modeling and survival analysis most pertinent to our proposed framework. Importantly, we state what properties our framework requires of the topic and survival models that will be combined to form a neural survival-supervised topic model. For ease of exposition, we phrase notation in terms of predicting time until death; other critical events are possible aside from death.

2.1 Data Format

We assume that we have access to a training dataset of n subjects, and we prespecify d historical clinical events to keep track of, where each event either occurs or not. For example, a clinical event could be whether a patient was ever diagnosed with diabetes up to present time. Continuous-valued clinical measurements could be discretized into bins to come up with such binary historical clinical events. For example, white blood count could be discretized

into five quintiles. Thus, one of the d events would then be "white blood count reading is in the bottom quintile"; this event could occur multiple times. For a given subject, we can count how many times each of the d events happened up to present time. We denote $X_{i,u}$ to be the number of times event $u \in \{1, \ldots, d\}$ occurred for subject $i \in \{1, \ldots, n\}$.[1] Viewing X as an n-by-d matrix, the i-th row of X (denoted by X_i) can be thought of as the feature vector for the i-th subject. Importantly, whether death has occurred is not one of the d historical events tracked by the matrix X since we will be predicting time until death.

As for the training label for the i-th subject, we have two recordings: indicator $\delta_i \in \{0, 1\}$ specifies whether death occurred for the i-th subject, and observed time $Y_i \in [0, \infty)$ is the i-th subject's "survival time" (time until death) if $\delta_i = 1$ or the "censoring time" if $\delta_i = 0$. The idea is that when we stop collecting training data, some subjects are still alive. The i-th subject still being alive corresponds to $\delta_i = 0$ with a true survival time that is unknown ("censored"); instead, we know that the subject's survival time is at least the censoring time.

2.2 Topic Modeling

Representing subjects using the matrix X above corresponds to topic modeling. Developed originally to analyze text [Blei et al., 2003], classically, a topic model represents each text document (in our case, each text document is a subject/patient) by raw counts of how many times d different "words" appear in the document (in our case, each word is a binary indicator for whether a past clinical event occurred). These raw counts are stored as the feature vector X_i described previously. A topic model transforms the i-th subject's feature vector X_i into a topic weight vector $W_i \in \mathbb{R}^k$, where $W_{i,g}$ measures how much of topic $g \in \{1, 2, \ldots, k\}$ is present in the i-th subject. A common assumption is that the i-th subject's feature vector W_i forms a probability distribution, i.e., the $W_{i,g} \geq 0$ for all words g and $\sum_{g=1}^{k} W_{i,g} = 1$. In the context of text documents, examples of topics include "sports", "finance", and "movies", so that a text document could be partially about both sports and finance but not movies, etc. In our case, topics could correspond, for example, to different patient age groups or having a specific severe illness. The goal is to automatically learn these topics.

[1] For simplicity, especially as the focus of our paper is not on feature engineering or preprocessing (which often needs to be tailored to specific datasets), when working with continuous-valued features, we use the simple quintile binning strategy we described along with counting how often each discretized event occurs across time to obtain the raw counts matrix X. In practice, one could of course use other discretization strategies, whether based on known threshold values that are already in clinical use for specific features, or based on automatically learned threshold values. Moreover, rather than counting how often a (discretized) measurement occurs over time, we could instead look at, for instance, the most recent value of that measurement, or the maximum value ever taken of that measurement across a time period, etc. Once again, choosing between these options could be done using existing clinical knowledge or learned automatically. We provide specific example approaches of how to discretize or summarize features over time in A.3, including taking advantage of recently developed machine learning methods. Importantly, our proposed framework accommodates any of these feature preprocessing strategies. We defer studying the effect of using different feature preprocessing strategies to future work.

As a concrete example of a topic model, we review the LDA model by Blei et al. [2003]. LDA assumes the topic weight vectors W_i's to be generated i.i.d. from a k-dimensional Dirichlet distribution. Next, to relate feature vector X_i to its topic weight vector W_i, let $\overline{X}_{i,u}$ denote the fraction of times a word appears for a specific subject, meaning that $\overline{X}_{i,u} = X_{i,u} / \left(\sum_{v=1}^{d} X_{i,v} \right)$. Then LDA assumes the factorization

$$\overline{X}_{i,u} = \sum_{g=1}^{k} W_{i,g} A_{g,u} \tag{2.1}$$

for a "topic-word" matrix $A \in \mathbb{R}^{k \times d}$, where each row of A is a distribution over the d vocabulary words; rows of A are assumed to be sampled i.i.d. from a d-dimensional Dirichlet distribution. Importantly, the different rows of A correspond to the different topics. Ideally each topic reveals words (or in our usage, historical clinical events) that are considered related or that tend to co-occur. A standard approach is, for example, to examine the most probable words per topic (i.e., identify the words with the highest values per row of A). We remark that equation (2.1) is commonly written compactly as the nonnegative matrix factorization $\overline{X} = WA$, where the matrix W has rows given by the different subjects' topic weight vectors W_i's.

Given matrix X, LDA estimates the matrices W and A (along with the parameters of the two Dirichlet distributions that generate rows of W and A) using variational inference (as done in the original paper by Blei et al. [2003]) or Gibbs sampling [Porteous et al., 2008]. Recently, Srivastava and Sutton [2017] showed how to approximate LDA in a neural net framework so that off-the-shelf neural net optimizers such as Adam [Kingma and Ba, 2014] can then be used to learn the model.

Prerequisites on the topic model for use with our framework Our proposed strategy for combining topic modeling with survival analysis can use any topic model with a neural net formulation that can output an estimate $\widehat{W}$ of the topic weight matrix W stated above. We shall feed $\widehat{W}$ as input to a survival model. We remark that our approach technically does not require the rows of W to be probability distributions, although as we show later, constraining W to be nonnegative can ease interpretation of the survival model used.

Aside from LDA, examples of neural topic models that can be used in our survival-supervised topic modeling framework include correlated topic models [Lafferty and Blei, 2006], supervised LDA [McAuliffe and Blei, 2008], SAGE [Eisenstein et al., 2011], ProdLDA [Srivastava and Sutton, 2017], and the Embedded Topic Model [Dieng et al., 2020]. As there are many neural topic models at this point, we refer the interested reader to the survey by Zhao et al. [2021].

2.3 Survival Analysis

Many standard topic models, including LDA, do not solve a prediction task. To predict time-to-event outcomes, we turn to survival analysis models. In this

section, we review some key concepts from survival analysis. More details can be found in standard textbooks (e.g., Kalbfleisch and Prentice 2002, Klein and Moeschberger 2006). At the end of this section, we state what our approach to combining topic and survival models requires of the survival model used.

Suppose we take the i-th subject's feature vector to be $W_i \in \mathbb{R}^k$ instead of X_i. As this notation suggests, when we combine topic and survival models, W_i corresponds to the i-th subject's topic weight vector; this strategy for combining topic and survival models was first done by Dawson and Kendziorski [2012], who extended the original supervised LDA formulation by McAuliffe and Blei [2008]. We treat the training data to the survival model as (W_1, Y_1, δ_1), $\ldots$, (W_n, Y_n, δ_n). Thus, the survival model does not get direct access to the "raw" feature vectors X_i's. Instead, it only gets information about the raw feature vectors through the topic weight vectors W_i's.

The prediction task The standard survival analysis prediction task can be stated as using the training data $(W_1, Y_1, \delta_1), \ldots, (W_n, Y_n, \delta_n)$ to estimate, for any test subject with feature vector $w \in \mathbb{R}^k$, the subject-specific survival function

$$S(t|w) = \mathbb{P}(\text{subject survives beyond time } t \mid \text{subject's feature vector is } w).$$

As with standard classification and regression settings, the training and test data are assumed to be i.i.d. samples from the same underlying distribution.

In survival analysis literature, often the prediction task is instead stated as estimating a transformed version of $S(\cdot|w)$ called the *hazard function*. Formally, let W_0 and T_0 be continuous random variables corresponding to the test subject's feature vector and the test subject's true survival time. We denote the cumulative distribution function (CDF) of T_0 given W_0 by $F(t|w) = \mathbb{P}(T_0 \leq t|W_0 = w)$, and the probability density function (PDF) of this distribution by $f(t|w) = \frac{\partial}{\partial t}F(t|w)$. The survival function is precisely $S(t|w) = 1 - F(t|w)$. The hazard function is

$$h(t|w) := -\frac{\partial}{\partial t}\log S(t|w) = \frac{-\frac{\partial}{\partial t}S(t|w)}{S(t|w)} = \frac{-\frac{\partial}{\partial t}[1 - F(t|w)]}{S(t|w)} = \frac{f(t|w)}{S(t|w)}, \quad (2.2)$$

which (from the right-most expression) is the instantaneous rate of death at time t divided by the probability of surviving up to time t, all conditioned on the feature vector being w. Given how the hazard function is defined, knowing $S(\cdot|w)$ means that we know $h(\cdot|w)$ and vice versa (i.e., if we know $h(\cdot|w)$, then $S(t|w) = \exp(-\int_0^t h(\tau|w)d\tau)$). Naturally, survival models differ in the assumptions they place on the underlying survival function $S(\cdot|w)$.

The technical challenge in estimating $S(\cdot|w)$ from training data is that in general, we do not observe the survival times for all of the training subjects: the observed times Y_i's are equal to survival times only for subjects who have $\delta_i = 1$; all other Y_i values are censoring times. We assume that the i-th training subject has survival time T_i and censoring time C_i that are conditionally independent given feature vector W_i, and if the survival time occurs before censoring

($T_i \leq C_i$), then $Y_i = T_i$ and $\delta_i = 1$; otherwise $Y_i = C_i$ and $\delta_i = 0$. This setup is referred to as *random censoring*.

Measuring survival prediction accuracy Although the prediction task can be described as estimating the survival function $S(\cdot|w)$ (or a variant of it such as the hazard function), when it comes to evaluating accuracy, we do not know the true function $S(\cdot|w)$ even in the training data. A number of evaluation metrics have been devised, for which we use the time-dependent concordance index C^{td} by Antolini et al. [2005]. Roughly, C^{td} measures the fraction of pairs of subjects correctly ordered by a survival model (based on estimated subject-specific survival functions) among pairs of subjects that can be unambiguously ordered. Thus, C^{td} scores are fractions between 0 and 1, and the highest accuracy corresponds to a value of 1.

Prerequisites on the survival model for use with our framework Our neural survival-supervised topic modeling framework requires that the survival model used can be learned by (sub)gradient descent using standard neural net optimizers. We will need to backpropagate through both the survival and topic models, which are linked via the topic weight matrix W (estimated by the topic model and treated as the input "feature vectors" by the survival model). Numerous survival models satisfy the criterion above of being learnable via (sub)gradient descent including the classical Cox proportional hazards model [Cox, 1972] and accelerated failure time (AFT) models (e.g., Cox 1972, Prentice 1978). We state the modeling assumptions of these models next along with their differentiable loss functions and how to construct an estimate $\widehat{S}(\cdot|w)$ for the subject-specific survival function $S(\cdot|w)$ after minimizing each model's loss function.

2.3.1 Example: Cox Proportional Hazards

The Cox model assumes that the hazard function has the form

$$h(t|w) = h_0(t)\exp(\beta^\top w) \qquad \text{for } t \geq 0,\ w \in \mathbb{R}^k, \tag{2.3}$$

where the two parameters are the baseline hazard function $h_0 : [0, \infty) \to [0, \infty)$, and the vector of regression coefficients $\beta \in \mathbb{R}^k$. Under random censoring (and actually more general censoring models), we can estimate β without knowing h_0 via maximizing a profile likelihood, which is equivalent to minimizing the differentiable loss function

$$L_{\text{Cox}}(\beta|W) = -\frac{1}{n}\sum_{i=1}^{n}\delta_i\Big[\beta^\top W_i - \log \sum_{j=1 \text{ s.t. } Y_j \geq Y_i}^{n} \exp(\beta^\top W_j)\Big]. \tag{2.4}$$

After computing parameter estimate $\widehat{\beta}$ by minimizing $L_{\text{Cox}}(\beta)$, we can estimate survival functions $S(\cdot|w)$ via the following approach by Breslow [1972]. Denote the unique times of death in the training data by $t_1, t_2, \ldots, t_m$. Let d_i be

the number of deaths at time t_i. We first compute the so-called hazard function $\widehat{h}_i := d_i / \left(\sum_{j=1 \text{ s.t. } Y_j \geq Y_i}^{n} \exp(\widehat{\beta}^\top W_j) \right)$ at each time index $i = 1, 2, \ldots, m$. Next, we form the "baseline" survival function $\widehat{S}_0(t) := \exp(-\sum_{i=1 \text{ s.t. } t_i \leq t}^{m} \widehat{h}_i)$. Finally, subject-specific survival functions are estimated to be powers of the baseline survival function: $\widehat{S}(t|w) := [\widehat{S}_0(t)]^{\exp(\widehat{\beta}^\top w)}$.

Importantly, under the Cox model, whether a subject with feature vector w is predicted to have overall higher or lower survival probabilities across time is determined by the inner product $\widehat{\beta}^\top w = \sum_{g=1}^{k} \widehat{\beta}_g w_g$. When this inner product is larger, then $\widehat{S}(t|w) = [\widehat{S}_0(t)]^{\exp(\widehat{\beta}^\top w)}$ is smaller across time. Recall that we shall take w to be a nonnegative topic weight vector, so the g-th topic being present for a subject means that $w_g > 0$. Note that the g-th topic's contribution to the inner product $\widehat{\beta}^\top w$ is precisely $\widehat{\beta}_g w_g$. Thus, the g-th topic having a larger $\widehat{\beta}_g$ coefficient means that the topic is associated with *lower* survival functions/probabilities, and thus *lower* mean (or median) survival times.[2] By ranking topics based on their $\widehat{\beta}_g$ values, we can thus get a sense of which topics are associated with lower vs higher survival times.

For the above loss $L_{\mathsf{Cox}}(\beta)$, we remark that one can regularize the Cox regression coefficients β. For example, adding a lasso, ridge, or more generally elastic-net penalty on β leads to the loss minimized by Simon et al. [2011]. Adding this regularization does not change how the hazard and survival functions are estimated once we have an estimate $\widehat{\beta}$ of β. Standard neural net optimizers can accommodate such a regularization term.

2.3.2 Example: Accelerated Failure Time Models

As another example of a survival model that our neural survival-supervised topic modeling framework can use, consider the log-logistic AFT model that assumes each subject's (possibly unobserved) survival time T_i has the form

$$\log T_i = \mu + \theta^\top W_i + \sigma \varepsilon_i, \tag{2.5}$$

where $\mu \in \mathbb{R}$, $\theta \in \mathbb{R}^k$, and $\sigma > 0$ are model parameters, and noise variables ε_i's are i.i.d. standard logistic, i.e., ε_i has PDF $f_\varepsilon(s) = e^s / \left(1 + e^s\right)^2$ and CDF $F_\varepsilon(s) = 1 / \left(1 + e^s\right)$. Thus, T_i given W_i is distributed as a log-logistic distribution and, in particular, the underlying survival function $S(\cdot|W_i)$ has a closed-form expression:

$$S(t|W_i) = \frac{1}{1 + t^{1/\sigma} \exp\{-(\mu + \theta^\top W_i)/\sigma\}} \qquad \text{for } t \geq 0. \tag{2.6}$$

[2] Note that the area under the survival function $\int_0^\infty S(t|w)dt$ is precisely the mean survival time for a subject with feature vector w. The time t for which $S(t|w)$ crosses $1/2$ is a median survival time for feature vector w. Thus, when the survival function decreases across all of time (except at time $t = 0$, where it is 1), then the mean and median survival times decrease.

Under random censoring, maximum likelihood estimation for μ, θ, and σ is equivalent to minimizing the differentiable loss function

$$L_{\mathsf{AFT}}(\theta, \mu, \sigma | W) := -\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_i \log f_\varepsilon(z_i) - \delta_i \log \sigma + (1 - \delta_i) \log \left(1 - F_\varepsilon(z_i)\right) \right\}, \tag{2.7}$$

where $z_i = (\log Y_i - \mu - \theta^\top W_i)/\sigma$. Hence, after minimizing the loss function $L_{\mathsf{AFT}}(\theta, \mu, \sigma | W)$, we have estimates $\widehat{\theta}$, $\widehat{\mu}$, and $\widehat{\sigma}$ for θ, μ, and σ respectively. We can plug these estimates into equation (2.6) to come up with an estimate $\widehat{S}(\cdot|w)$ for any feature vector w.

Interpretation of the log-logistic AFT model is similar to that of the Cox model. As we take the feature vector w to be a topic weight vector with nonnegative values, once again whether the predicted survival function has higher or lower probabilities is determined by an inner product, this time $\widehat{\theta}^\top w$. However, unlike in the Cox model, where the g-th topic having larger Cox regression coefficient $\widehat{\beta}_g$ means that the g-th topic is associated with *shorter* mean/median survival times, for the above AFT model, having larger regression coefficient $\widehat{\theta}_g$ means that the g-th topic is associated with *longer* mean/median survival times.[3]

Other AFT models are also possible where, for example, T_i given W_i has a log-normal, Weibull, gamma, generalized gamma, or inverse-Gaussian distribution instead of a log-logistic distribution. These different models arise from changing the distribution of the i.i.d. noise terms ε_i's in equation (2.5). Moreover, just as with the Cox model, we could introduce regularization.

As stated previously, in this paper we use the time-dependent concordance index accuracy metric, which is based on ranking pairs of subjects. As such, using a ranking-based regularization term when learning a survival model tends to yield higher c-index values, which has been previously reported by other researchers (e.g., Chapfuwa et al. 2018, Lee et al. 2018, Kvamme et al. 2019). Accounting for these previous researchers' findings, in our experiments later when we use an AFT model, we use the same regularization strategy as Chapfuwa et al. [2018] by adding the ranking loss by Steck et al. [2007]:

$$L_{\mathsf{ranking}}(\theta) = -1 + \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \log_2 \left\{ 1 + \exp\left(\theta^\top (W_i - W_j)\right) \right\}, \tag{2.8}$$

where $\mathcal{E}$ consists of pairs of subjects (i, j) such that $\delta_i = 1$ (death is observed for the i-th training subject) and moreover $Y_j > Y_i$ (the observed time for the j-th training subject is higher than that of the i-th subject). Steck et al. [2007] show that $-L_{\mathsf{ranking}}(\theta)$ is a lower bound on a variant of concordance index; thus, minimizing $L_{\mathsf{ranking}}(\theta)$ aims to maximize concordance index. Note that

[3] Under the log-logistic AFT model, the median survival time for a subject with feature vector w is $\exp(\mu + \theta^\top w)$. The mean survival time exists only if $\sigma < 1$ for which it is given by $\frac{\pi\sigma \exp(\mu + \theta^\top w)}{\sin(\pi\sigma)}$.

the Cox model does not need a ranking regularizer since it already approximately maximizes concordance index [Steck et al., 2007].

Importantly, in how we combine neural topic models with survival analysis, for the resulting overall model to be readily interpretable, choosing a simple interpretable survival model is crucial, as we have illustrated with the above Cox and log-logistic AFT examples. Thus, although our approach is indeed compatible with survival models given by deep neural net extensions of Cox and AFT models (e.g., Faraggi and Simon 1995, Katzman et al. 2018, Chapfuwa et al. 2018, Kvamme et al. 2019, Kvamme and Borgan 2021) that can be more accurate at time-to-event predictions than classical non-neural-net methods and that can learn highly nonlinear functions of the input feature vector, these deep survival models are typically difficult to interpret.

3 Neural Survival-Supervised Topic Models

We now present our proposed neural survival-supervised topic modeling framework that can combine any neural topic model and any survival model meeting the prerequisites stated in Sections 2.2 and 2.3. For ease of exposition, we first explain how to combine LDA with the Cox proportional hazards model, similar to what is done by Dawson and Kendziorski [2012] except we do this combination in a neural net framework. To show the flexibility of our framework, we explain how to combine LDA with the log-logistic AFT model, and how to replace LDA with the SAGE topic model.

3.1 A Neural Formulation of the LDA/Cox Combination

We first need a neural net formulation of LDA. We can use the SCHOLAR framework by Card et al. [2018]. Card et al. do not explicitly consider survival analysis in their setup although they mention that predicting different kinds of real-valued outputs can be incorporated by using different label networks. We use their same setup and have the final label network perform survival analysis. We give an overview of SCHOLAR before explaining our choice of label network. Note that for clarity of presentation, we present a slightly simplified version of SCHOLAR.

The SCHOLAR framework specifies a generative model for the data, including how each individual word in each subject is generated. In particular, recall that $X_{i,u}$ denotes the number of times the word $u \in \{1, 2, \ldots, d\}$ appears for the i-th subject. Let v_i denote the number of words for the i-th subject, i.e., $v_i = \sum_{u=1}^{d} X_{i,u}$. We now define the random variable $\psi_{i,\ell} \in \{1, 2, \ldots, d\}$ to be what the ℓ-th word for the i-th subject is (for $i = 1, 2, \ldots, n$ and $\ell = 1, 2, \ldots, v_i$). Then the generative process for SCHOLAR with k topics is as follows, stated for the i-th subject:

1. Generate the i-th subject's topic distribution:

(a) Sample $\widetilde{W}_i \sim \mathcal{N}(\boldsymbol{\mu}_0, \text{diag}(\boldsymbol{\sigma}_0^2))$, where $\boldsymbol{\mu}_0 \in \mathbb{R}^k$ and $\boldsymbol{\sigma}_0^2 \in [0, \infty)^k$ are user-specified, and $\text{diag}(\cdot)$ constructs a diagonal matrix from a vector.

(b) Set the i-th subject's topic weights vector to be $W_i = \text{softmax}(\widetilde{W}_i)$.

2. Generate the i-th subject's words:

   (a) Compute the i-th subject's word distribution $\phi_i = f_{\mathsf{word}}(W_i)$, where f_{word} is a generator network.

   (b) For word $\ell = 1, 2, \ldots, v_i$: Sample $\psi_{i,\ell} \sim \text{Multinomial}(\phi_i)$.

3. Generate the i-th subject's output label:

   Sample Y_i from a distribution parameterized by label network $f_{\mathsf{label}}(W_i)$.

Different choices for the parameters $\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2$, f_{word}, and f_{label} lead to different topic models. To approximate LDA where topic distributions are sampled from a symmetric Dirichlet distribution with parameter $\alpha > 0$, we set $\boldsymbol{\mu}_0$ to be the all zeros vector, $\boldsymbol{\sigma}_0^2$ to have all entries equal to $(k-1)/(\alpha k)$, and $f_{\mathsf{word}}(w) = w^\top A$, where $A \in \mathbb{R}^{k \times d}$ has a Dirichlet prior per row; in fact the matrix A is the same as the one in equation (2.1). Standard LDA is unsupervised so step 3 of the above generative process would be omitted. In terms of implementation, we set the g-th row of A to be $A_g = \text{softmax}(H_g)$ for an unconstrained matrix $H \in \mathbb{R}^{k \times d}$, and for simplicity, we assume the prior on each row of A to be uniform (a special case of a Dirichlet prior).

3.1.1 Learning Topic Model Parameters

The topic model parameters are learned via amortized variational inference [Kingma and Welling, 2014, Rezende et al., 2014]. We summarize this procedure for the above unsupervised LDA neural net approximation including stating the loss function. For the derivation of this procedure and loss function, see Section 3.2 of Card et al. [2018].

For the i-th subject, we keep track of a distribution $q_i := \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2))$, where $\boldsymbol{\mu}_i \in \mathbb{R}^k$ and $\boldsymbol{\sigma}_i^2 \in [0, \infty)^k$ will be defined shortly. Distribution q_i approximates the posterior of unnormalized topic weights $\widetilde{W}_i$ given the observed words $\psi_i := (\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,v_i})$. We introduce a multilayer perceptron $f_e : \mathbb{R}^d \to \mathbb{R}^{d'}$ that takes as input X_i (the word counts for the i-th subject) and outputs an embedding $\boldsymbol{\pi}_i = f_e(X_i)$, where the embedding dimension d' is user-specified. Then we set

$$\boldsymbol{\mu}_i = \mathbf{W}_\mu \boldsymbol{\pi}_i + \boldsymbol{b}_\mu, \tag{3.1}$$

$$\log(\boldsymbol{\sigma}_i^2) = \mathbf{W}_\sigma \boldsymbol{\pi}_i + \boldsymbol{b}_\sigma. \tag{3.2}$$

The variables $\mathbf{W}_\mu \in \mathbb{R}^{d' \times k}$, $\boldsymbol{b}_\mu \in \mathbb{R}^k$, $\mathbf{W}_\sigma \in \mathbb{R}^{d' \times k}$, and $\boldsymbol{b}_\sigma \in \mathbb{R}^k$ are parameters. In the latter equation, log is applied element-wise. In summary, the model

parameters we aim to learn are $\mathbf{W}_\mu$, $\boldsymbol{b}_\mu$, $\mathbf{W}_\sigma$, and $\boldsymbol{b}_\sigma$, the parameters for the multilayer perceptron f_e, and finally the matrix H (recall that for LDA, we set $f_{\mathsf{word}}(w) = w^\top A$ with $A_g = \text{softmax}(H_g)$ in step 2 of the SCHOLAR generative process). We collectively refer to all the parameters as $\boldsymbol{\Theta}_{\mathsf{LDA}}$. Meanwhile, the number of topics k, constant $\alpha > 0$ (used in the Dirichlet prior for unnormalized topic weights), and the neural architecture of f_e are hyperparameters that are user-specified.

As is standard now in amortized variational inference, the loss function is randomly computed given parameters $\boldsymbol{\Theta}_{\mathsf{LDA}}$; hyperparameters and the input raw counts matrix X are treated as fixed. For the i-th subject, we sample an unnormalized topic weight vector $\widetilde{W}_i^{(\mathsf{s})} \sim q_i$. Then following steps 1(b) and 2(a) of the SCHOLAR generative process, we compute the topic weight vector $W_i^{(\mathsf{s})} = \text{softmax}(\widetilde{W}_i^{(\mathsf{s})})$ and word distribution $\zeta_i^{(\mathsf{s})} := W_i^{(\mathsf{s})\top} A \in [0,1]^d$. We repeat this across all subjects i. Then the loss function minimized by SCHOLAR for LDA is

$$\widetilde{L}_{\mathsf{LDA}}(\boldsymbol{\Theta}_{\mathsf{LDA}}) = -\frac{1}{n}\sum_{i=1}^{n}\Bigg[\overbrace{\sum_{v=1}^{d} X_{i,v}\log(\zeta_{i,v}^{(\mathsf{s})})}^{\text{log likelihood of observed words}} \underbrace{-\frac{1}{2}\sum_{g=1}^{k}\Big(\frac{\boldsymbol{\sigma}_{i,g}^2 + \boldsymbol{\mu}_{i,g}^2}{(k-1)/(\alpha k)} - k + \log\frac{(k-1)/(\alpha k)}{\boldsymbol{\sigma}_{i,g}^2}\Big)}_{\text{KL divergence between } q_i \text{ and true posterior}}\Bigg]. \tag{3.3}$$

When we apply this framework to clinical data, one practical issue is that some subjects have dramatically more historical clinical measurements than other subjects. For example, in one dataset in our experiments, one subject has a total of 59824 measurements (note that the same "word"/past historical clinical event could occur multiple times) whereas there is another subject who has a total of 3 measurements! When there is such heterogeneity in how many words are present per "document"/subject, the subjects with a very large number of historical clinical measurements will dominate the entire loss function above. To prevent this behavior, for all datasets, we replace the raw word counts X with its normalized version $\overline{X}$ stated in Section 2.2 ($\overline{X}$ is obtained by taking X and dividing each row by the sum of the row), which effectively weights every subject equally (despite subjects possibly having varying amounts of measure-

ments present).[4] Thus, the loss function we minimize is instead

$$L_{\mathsf{LDA}}(\boldsymbol{\Theta}_{\mathsf{LDA}}) = -\frac{1}{n}\sum_{i=1}^{n}\Bigg[\sum_{v=1}^{d}\overline{X}_{i,v}\log(\zeta_{i,v}^{(\mathsf{s})}) - \frac{1}{2}\sum_{g=1}^{k}\Big(\frac{\boldsymbol{\sigma}_{i,g}^2+\boldsymbol{\mu}_{i,g}^2}{(k-1)/(\alpha k)} - k + \log\frac{(k-1)/(\alpha k)}{\boldsymbol{\sigma}_{i,g}^2}\Big)\Bigg]. \tag{3.4}$$

We can minimize this loss with respect to $\boldsymbol{\Theta}_{\mathsf{LDA}}$ using standard neural net optimizers as well as train in minibatches to scale to large datasets. Empirically, Srivastava and Sutton [2017] and Card et al. [2018] have found that for training neural topic models, training with high momentum and using batch normalization is essential in preventing the topics learned from being the same (the so-called issue of "mode collapse"); for the interested reader, see the implementation notes in Appendix C of Card et al. [2018].

Recall from Section 2.2 that we require the neural topic model used in our framework to be able to output estimated topic weight vectors $\widehat{W}_i$'s for the different subjects as these will be used as inputs to the survival model. We could simply set $\widehat{W}_i$ to be the topic weight vector $W_i^{(\mathsf{s})} = \mathrm{softmax}(\widetilde{W}_i^{(\mathsf{s})})$ constructed based on the random unnormalized topic weight vector $\widetilde{W}_i^{(\mathsf{s})} \sim q_i$. Alternatively, rather than only using one sample $\widetilde{W}_i^{(\mathsf{s})}$, we could draw multiple samples $\widetilde{W}_i^{(\mathsf{s}),1}, \ldots, \widetilde{W}_i^{(\mathsf{s}),\ell} \overset{\text{i.i.d.}}{\sim} q_i$, and output $\widehat{W}_i = \frac{1}{\ell}\sum_{j=1}^{\ell}\mathrm{softmax}(\widetilde{W}_i^{(\mathsf{s}),j})$.

3.1.2 Survival Supervision

To incorporate the Cox survival loss, we set step 3 of the SCHOLAR generative process to use $f_{\mathsf{label}}(W_i) = \beta^\top W_i$ for parameter vector $\beta \in \mathbb{R}^k$, where we explicitly constrain $\beta_k = 0$, i.e., how much of the k-th topic is present is ignored in the inner product calculation. This is done so that the k-th topic acts as a background topic. We remark that $f_{\mathsf{label}}(W_i)$ is simple to implement: given W_i, we drop the entry corresponding to the k-th topic and then feed the result to a standard linear layer with a single output node and no bias term. The weights of this fully-connected layer thus correspond to $(\beta_1, \beta_2, \ldots, \beta_{k-1})$. The last coefficient $\beta_k = 0$ is not stored.

Note that β precisely consists of the Cox regression coefficients in equation (2.3). Meanwhile, $f_{\mathsf{label}}(W_i)$ precisely takes the role of the $\beta^\top W_i$ terms in the Cox loss (2.4). Of course, as we do not observe the true topic weight vector

[4] Other approaches are possible for weighting different subjects. For instance, instead of using the row-normalized matrix $\overline{X}$ or the raw counts matrix X, we could interpolate between these two choices by using $\overline{X}_{i,u}^{(\xi)} := X_{i,u}/(\sum_{v=1}^{d} X_{i,v})^{\xi}$, where $\xi \in [0,1]$ is a user-specified hyperparameter (setting $\xi = 1$ corresponds to using $\overline{X}_{i,u}$, whereas setting $\xi = 0$ corresponds to using the raw count $X_{i,u}$). For simplicity, we simply use $\overline{X}$ in our experiments later.

W_i, we plug in its estimate $\widehat{W}_i$ from the topic model. To summarize, the Cox loss we use with the neural topic model is

$$L_{\text{Cox-with-background-topic}}(\beta_1, \ldots, \beta_{k-1} | \widehat{W}) = -\frac{1}{n} \sum_{i=1}^{n} \delta_i \Big[f_{\text{label}}(\widehat{W}_i) - \log \sum_{j=1 \text{ s.t. } Y_j \geq Y_i}^{n} \exp(f_{\text{label}}(\widehat{W}_j)) \Big], \tag{3.5}$$

where we have left out regression coefficient β_k as it is constrained to be 0.

We can now state the overall loss function that we minimize for the neural LDA-Cox model:

$$L_{\text{LDA-Cox}}(\mathbf{\Theta}_{\text{LDA}}, \beta_1, \ldots, \beta_{k-1}) = L_{\text{LDA}}(\mathbf{\Theta}_{\text{LDA}}) + \lambda_{\text{survival}} L_{\text{Cox-with-background-topic}}(\beta_1, \ldots, \beta_{k-1} | \widehat{W}), \tag{3.6}$$

where hyperparameter $\lambda_{\text{survival}} > 0$ weights the importance of the survival loss. We refer to the resulting model as SCHOLAR LDA-COX.

3.1.3 Model Interpretation

For the g-th topic learned, we can look at its distribution over words $A_g \in [0,1]^d$ (the g-th row of A given in equation (2.1)) and, for instance, rank words by their probability of appearing for topic g. The g-th topic is also associated with Cox regression coefficient β_g, where each β_g is the parameter from equation (3.5). Again, the k-th topic is constrained to have Cox regression coefficient $\beta_k = 0$. Under the Cox model, β_g being larger means that the g-th topic is associated with *shorter* mean/median survival times, as discussed in Section 2.3.1.

3.2 Using Other Choices of Topic or Survival Models

To give a sense of the generality of our proposed framework, we explain how to derive neural survival-supervised topic models corresponding to combining LDA with an AFT model (Section 3.2.1) as well as combining the SAGE topic model [Eisenstein et al., 2011] with either Cox or AFT survival models (Section 3.2.2).

3.2.1 LDA/AFT

To combine LDA with an AFT survival model, we use the same idea as how we combined LDA with a Cox model. The changes are as follows. First off, in step 3 of the SCHOLAR generative process, we now set $f_{\text{label}}(W_i) = \theta^\top W_i + \mu$, again constraining the k-th regression coefficient $\theta_k = 0$ to correspond to a background topic. Effectively, we are taking the survival time T_i to be of the form $\log T_i = f_{\text{label}}(W_i) + \sigma\varepsilon_i$ in equation (2.7), where parameters μ, θ, and σ are the same as described in Section 2.3.2 except with the new constraint that $\theta_k = 0$.

Note that $f_{\text{label}}(W_i)$ can be implemented by taking the input W_i, dropping the k-th topic's weight, and then feeding the result through a standard linear layer with one output node and a bias term. The bias term is precisely μ and the weight matrix of the linear layer precisely gives $(\theta_1, \theta_2, \ldots, \theta_{k-1})$. As the true W_i is unknown, we plug in its estimate $\widehat{W}_i$ from the topic model.

We use the regularized survival loss function

$$\begin{aligned} & L_{\text{AFT-with-background-topic}}(\mu, \sigma, \theta_1, \ldots, \theta_{k-1} | \widehat{W}) \\ & \quad = -\frac{1}{n} \sum_{i=1}^{n} \left\{ \delta_i \log f_\varepsilon(Z_i) - \delta_i \log \sigma + (1 - \delta_i) \log \left(1 - F_\varepsilon(Z_i)\right) \right\} \\ & \qquad + \lambda_{\text{ranking}} L_{\text{ranking}}(\theta_1, \ldots, \theta_{k-1}), \end{aligned} \tag{3.7}$$

where $Z_i = [(\log(Y_i)) - f_{\text{label}}(\widehat{W}_i)]/\sigma$, $f_\varepsilon(s) = e^s/\left(1 + e^s\right)^2$, $F_\varepsilon(s) = 1/\left(1 + e^s\right)$, and $\lambda_{\text{ranking}} \geq 0$ is a user-specified hyperparameter, and $L_{\text{ranking}}(\theta_1, \ldots, \theta_{k-1})$ is the same as in equation (2.8) except with the constraint $\theta_k = 0$. Since parameter σ needs to be strictly positive, we instead have the neural net keep track of $\log \sigma$, which is unconstrained and we initialize with a random sample from $\mathcal{N}(0, 10^{-4})$. The overall loss to be minimized is thus

$$\begin{aligned} & L_{\text{LDA-AFT}}(\boldsymbol{\Theta}_{\text{LDA}}, \mu, \sigma, \theta_1, \ldots, \theta_{k-1}) \\ & \quad = L_{\text{LDA}}(\boldsymbol{\Theta}_{\text{LDA}}) + \lambda_{\text{survival}} L_{\text{AFT-with-background-topic}}(\mu, \sigma, \theta_1, \ldots, \theta_{k-1} | \widehat{W}), \end{aligned} \tag{3.8}$$

for a user-specified hyperparameter $\lambda_{\text{survival}} > 0$. The rest of neural net training works exactly the same way as in the LDA-Cox combination.

As for model interpretation, just as with the LDA-Cox model, for the g-th topic, we can inspect its distribution over words given by the g-th row of the matrix A. As discussed in Section 2.3.2, the g-th topic has an associated regression coefficient θ_g for which larger values mean that the g-th topic is associated with longer mean/median survival times.

3.2.2 Replacing LDA with SAGE

The above LDA/Cox and LDA/AFT combinations can easily accommodate replacing LDA with a different neural topic model. For example, to replace LDA with SAGE [Eisenstein et al., 2011], we make the following changes. First, recall that in step 2(a) of the SCHOLAR generative process, the neural net f_{word} maps an input topic weight vector w to a distribution over d words. For SAGE, we set f_{word} to be

$$f_{\text{word}}(w) = \text{softmax}(\gamma + w^\top H),$$

where $\gamma \in \mathbb{R}^d$ and $H \in \mathbb{R}^{k \times d}$ are parameters. Note that in a neural net framework, f_{word} is implemented as a linear layer followed by softmax activation. Specifically, the linear layer has a bias term and maps feature vectors of size k to output vectors of size d. The linear layer's weight matrix and bias term correspond to H and γ, respectively.

The interpretation is as follows: given a subject with topic weight vector w, the v-th word (a historical clinical event) occurs with probability proportional to $\exp(\gamma_v + \sum_{g=1}^{k} w_g H_{g,v})$. In this sense, γ_v can be thought of as a background log frequency of the v-th word. The g-th topic is then represented by the g-th row of H and can be thought of as log deviations from the background log frequency vector. Phrased informally, SAGE represents each topic as a deviation from background word frequencies. This representation is convenient in that there often are many "background" words that appear in a very large fraction of subjects and are not helpful in distinguishing between the topics. For LDA, these background words would have to be removed either as a preprocessing or as a postprocessing step. SAGE on the other hand inherently accounts for these background words.

For SAGE, to interpret the g-th topic, we can rank words the words from largest to smallest deviation from background according to the values in the g-th row of H. The values are of course not probabilities. For example, for the g-th topic, if the v-th word has a log deviation value $H_{g,v} = 3$, then it means that it occurs $\exp(3)$ times more than word v's background frequency. It is of course possible to have negative log deviation values.

The loss function we use to learn the SAGE topic model is almost the same as for LDA and is given by

$$\begin{aligned} L_{\mathsf{SAGE}}(\boldsymbol{\Theta}_{\mathsf{SAGE}}) = -\frac{1}{n}\sum_{i=1}^{n}\Bigg[\sum_{v=1}^{d}\overline{X}_{i,v}\log(\zeta_{i,v}^{(\mathbf{s})}) \\ - \frac{1}{2}\sum_{g=1}^{k}\Big(\frac{\boldsymbol{\sigma}_{i,g}^2 + \boldsymbol{\mu}_{i,g}^2}{(k-1)/(\alpha k)} - k + \log\frac{(k-1)/(\alpha k)}{\boldsymbol{\sigma}_{i,g}^2}\Big)\Bigg] \\ + \lambda_{\text{small-deviation}}\sum_{g=1}^{k}\sum_{v=1}^{d}H_{g,v}^2, \end{aligned} \tag{3.9}$$

where the differences are that: (a) we redefine $\zeta_i^{(\mathbf{s})} = \text{softmax}(\gamma + W_i^{(\mathbf{s})\top} H)$, and (b) we add an ℓ_2 regularization term on the log deviations, with a user-specified weight $\lambda_{\text{small-deviation}} \geq 0$. The rest of the setup is the same as for LDA, and we collectively denote the complete set of parameters that we minimize the loss over as $\boldsymbol{\Theta}_{\mathsf{SAGE}}$. By combining this topic model with the Cox and log-logistic AFT survival models, we obtain SCHOLAR SAGE-COX and SCHOLAR SAGE-AFT.

We remark that the original SAGE model actually also uses ℓ_1 regularization on the log deviations in H, but in preliminary experiments, we found that encouraging sparsity yields topic models that are not clinically interpretable. The issue is that in healthcare, often times, a collection of clinical measurements help explain a condition. When these measurements are collinear or have high pairwise correlation, enforcing sparsity would favor just retaining one of these measurements and zeroing out the contributions of the others [Zou and Hastie, 2005, Section 2.3]. Consequently, we lose valuable co-occurrence information

Table 2: Basic characteristics of the survival datasets used.

Dataset	Description	Number of subjects	Number of features	Fraction censored
SUPPORT1	acute respiratory failure/multiple organ system failure	4194	14	35.6%
SUPPORT2	COPD/congestive heart failure/cirrhosis	2804	14	38.8%
SUPPORT3	cancer	1340	13	11.3%
SUPPORT4	coma	591	14	18.6%
METABRIC	breast cancer	1981	24	55.2%
UNOS	heart transplant	62644	49	50.2%
MIMIC-ICH	intracerebral hemorrhage	961	1530	23.1%

of related clinical features. For this reason, as well as the previous empirical finding by Card et al. [2018] that encouraging sparsity results in worse topics learned in terms of other standard topic modeling metrics of perplexity and coherence, we do not encourage sparsity in learning the topic log deviations matrix H.

4 Experiments

4.1 Data

We conduct experiments on seven datasets: data on severely ill hospitalized patients from the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) [Knaus et al., 1995], which—as suggested by Harrell [2015]—we split into four datasets corresponding to different disease groups (acute respiratory failure/multiple organ system failure, cancer, coma, COPD/congestive heart failure/cirrhosis); data from breast cancer patients (METABRIC) [Curtis et al., 2012]; data from patients who received heart transplants in the United Network for Organ Sharing (UNOS);[5] and lastly patients with intracerebral hemorrhage (ICH) from the MIMIC-III electronic heath records dataset [Johnson et al., 2016]. For all except the last dataset, we predict time until death; for the ICH patients, we predict time until discharge from a hospital ICU. Basic characteristics of these datasets are reported in Table 2. More details on the datasets and on data preproprocessing are in A. We randomly divide each dataset into a 80%/20% train/test split.

4.2 Experimental Setup

We benchmark SCHOLAR LDA-COX, SCHOLAR LDA-AFT, SCHOLAR SAGE-COX, and SCHOLAR SAGE-AFT against 5 baselines:

[5]We use the UNOS Standard Transplant and Analysis Research data from the Organ Procurement and Transplantation Network as of September 2019, requested at: `https://www.unos.org/data/`

- 2 classical methods (lasso-regularized Cox [Simon et al., 2011], and random survival forests (RSF) [Ishwaran et al., 2008])
- 2 deep learning methods (DeepSurv [Katzman et al., 2018] and DeepHit [Lee et al., 2018])
- a naive two-stage decoupled LDA/Cox model (fit unsupervised LDA first and then fit a Cox model)

For all methods, we hold out 20% of the training data as a validation set to select hyperparameters. Hyperparameter search grids are included in B. For evaluating a model's prediction accuracy on the validation set as well as the final test set, we use the time-dependent concordance C^{td} index [Antolini et al., 2005]. For every test set C^{td} index reported, we also compute its 95% confidence interval, which we obtain by taking 1000 bootstrap samples of the test set with replacement, recomputing the C^{td} index per bootstrap sample, and taking the 2.5 and 97.5 percentile values among the C^{td} indices computed.

4.3 Results

Test set C^{td} indices are reported in Table 3 with the 95% bootstrap confidence intervals. The main takeaways are that:

1. Random survival forest is clearly a strong baseline for the datasets considered, often outperforming the deep learning baselines DEEPSURV and DEEPHIT. That said, no single model is consistently the best.
2. The different neural survival-supervised topic models tested have accuracy scores that are often quite similar with each other.
3. The neural survival-supervised topic models often achieve accuracy scores as good as deep neural net baselines. For example, if we ignore the confidence intervals for a moment and go by test set C^{td} index alone, SCHOLAR LDA-COX's accuracy scores on SUPPORT3, UNOS, and MIMIC-ICH are better than those of DEEPSURV. Meanwhile, SCHOLAR LDA-COX's accuracy scores on SUPPORT1, SUPPORT3, METABRIC, UNOS, MIMIC-ICH are better than those of DEEPHIT. However, the differences are often small and, especially once we account for the confidence intervals, we would not claim that neural survival-supervised topic models yield more accurate predictions than the deep learning baselines or vice versa.
4. Clearly, the naive approach (NAIVE LDA-COX) of fitting an unsupervised topic model first and then separately training a Cox model using the topics learned tends to achieve worse accuracy scores than its supervised counterpart SCHOLAR LDA-COX.

To supplement our third takeaway above, specifically for SCHOLAR LDA-COX, we also use bootstrap sampling to compute differences between C^{td} indices of SCHOLAR LDA-COX vs different baseline models. Specifically, we repeat the

Table 3: Test set C^{td} indices with 95% bootstrap confidence intervals.

Model	Dataset						
	SUPPORT1	SUPPORT2	SUPPORT3	SUPPORT4	METABRIC	UNOS	MIMIC-ICH
COX	0.632 (0.609, 0.658)	0.557 (0.523, 0.592)	**0.581** (0.543, 0.617)	0.508 (0.437, 0.578)	0.675 (0.630, 0.715)	0.594 (0.585, 0.602)	0.612 (0.551, 0.659)
RSF	**0.658** (0.632, 0.685)	0.578 (0.545, 0.609)	0.558 (0.516, 0.601)	0.547 (0.480, 0.612)	**0.712** (0.670, 0.755)	**0.604** (0.596, 0.612)	0.618 (0.567, 0.666)
DEEPSURV	0.649 (0.625, 0.673)	0.568 (0.535, 0.600)	0.556 (0.515, 0.597)	0.538 (0.466, 0.606)	0.706 (0.667, 0.745)	0.597 (0.588, 0.604)	0.615 (0.565, 0.667)
DEEPHIT	0.633 (0.606, 0.658)	0.563 (0.531, 0.596)	0.564 (0.526, 0.603)	0.516 (0.449, 0.583)	0.666 (0.620, 0.710)	0.585 (0.576, 0.593)	0.587 (0.533, 0.637)
NAIVE LDA-COX	0.602 (0.577, 0.626)	0.544 (0.512, 0.578)	0.515 (0.475, 0.555)	**0.554** (0.485, 0.621)	0.639 (0.589, 0.686)	0.540 (0.532, 0.549)	0.537 (0.484, 0.591)
SCHOLAR LDA-COX	0.637 (0.612, 0.663)	0.560 (0.527, 0.591)	0.569 (0.533, 0.607)	0.510 (0.439, 0.572)	0.696 (0.653, 0.737)	0.600 (0.591, 0.608)	**0.639** (0.588, 0.687)
SCHOLAR LDA-AFT	0.632 (0.607, 0.657)	**0.586** (0.554, 0.617)	0.551 (0.512, 0.591)	0.529 (0.457, 0.599)	0.688 (0.643, 0.728)	0.596 (0.588, 0.604)	0.634 (0.585, 0.680)
SCHOLAR SAGE-COX	0.605 (0.580, 0.631)	0.558 (0.526, 0.593)	0.560 (0.522, 0.598)	0.470 (0.405, 0.529)	0.708 (0.669, 0.746)	0.603 (0.595, 0.611)	0.629 (0.579, 0.677)
SCHOLAR SAGE-AFT	0.635 (0.611, 0.660)	0.550 (0.516, 0.583)	0.564 (0.526, 0.600)	0.550 (0.484, 0.621)	0.700 (0.659, 0.742)	0.599 (0.591, 0.606)	0.631 (0.579, 0.681)

following 1000 times: (a) take a bootstrap sample from the test set, (b) compute the bootstrap sample's predictions using SCHOLAR LDA-COX as well as a baseline model, (c) compute the C^{TD} index of SCHOLAR LDA-COX's predictions minus that of the baseline model's predictions. Thus, we have 1000 differences in C^{TD} indices, for which we then take the 2.5 and 97.5 percentiles to get a 95% confidence interval. We report these confidence intervals in Table 4. We find that 0 is in all the confidence intervals for SCHOLAR LDA-COX vs DEEPSURV and nearly in all the ones for SCHOLAR LDA-COX vs DEEPHIT (in fact, the only times 0 is not included for DEEPHIT is for the UNOS and MIMIC-ICH datasets, in which SCHOLAR LDA-COX is more accurate). We omit tables that compare the other neural survival-supervised topic models with various baselines as they follow similar trends. To reiterate, we do not claim that our proposed models outperform the various baselines tested. Instead we claim that they achieve accuracy that is competitive with deep learning baselines. In fact, Tables 3 and 4 suggest that SCHOLAR LDA-COX is competitive with COX and RSF as well. On the other

Table 4: 95% bootstrap confidence intervals for the test set C^{td} index of SCHOLAR-LDA minus that of various baselines (when this difference is positive, it means that SCHOLAR-LDA is more accurate than a particular baseline).

Baseline	Dataset						
	SUPPORT1	SUPPORT2	SUPPORT3	SUPPORT4	METABRIC	UNOS	MIMIC-ICH
COX	(-0.010, 0.018)	(-0.024, 0.026)	(-0.045, 0.024)	(-0.073, 0.074)	(-0.015, 0.059)	(0.002, 0.010)	(-0.029, 0.088)
RSF	(-0.038, -0.006)	(-0.050, 0.014)	(-0.026, 0.047)	(-0.103, 0.032)	(-0.041, 0.009)	(-0.010, -0.000)	(-0.027, 0.070)
DEEPSURV	(-0.029, 0.004)	(-0.042, 0.026)	(-0.020, 0.046)	(-0.098, 0.044)	(-0.039, 0.019)	(-0.002, 0.009)	(-0.010, 0.059)
DEEPHIT	(-0.016, 0.024)	(-0.035, 0.030)	(-0.041, 0.051)	(-0.095, 0.081)	(-0.006, 0.069)	(0.007, 0.024)	(0.010, 0.100)
NAIVE LDA-COX	(0.013, 0.058)	(-0.020, 0.054)	(0.013, 0.096)	(-0.133, 0.036)	(0.028, 0.088)	(0.053, 0.066)	(0.031, 0.170)

hand, the NAIVE LDA-COX baseline does appear to be significant less accurate than SCHOLAR LDA-COX for all datasets except SUPPORT2 and SUPPORT4.

4.4 Interpretability of Baselines

Importantly, we remark that the deep learning baselines DEEPSURV and DEEPHIT do not produce interpretable models and they were not designed to be interpretable. Random survival forests are also not easily interpretable: while a single decision tree could be interpretable if its depth and number of leaves are not too large, the difficulty in interpreting a learned random survival forest model is that there are many trees (in our experiments, we use 100 trees for each model), and the best-performing models tend to have learned trees that are moderate in size (e.g., a depth of 6 with 64 leaves). Having to look at 100 moderate-sized trees to interpret a single random survival forest model is not that simple, and it is not straightforward teasing apart how features are related without instead using some post hoc explanation approach like SHAP [Lundberg and Lee, 2017] or TreeExplainer [Lundberg et al., 2020]. Of the models evaluated, only the Cox model and the survival-supervised topic models can readily be interpreted. However, as mentioned in Section 1, Cox models do not inherently learn how features relate, and one would have to introduce new features that encode interactions, which becomes impractical when the number of features is large.

4.5 Interpretability of Neural Survival-Supervised Topic Models

We next discuss interpretability of neural survival-supervised topic models. As there are many models considered, for ease of exposition, we only present

results for SCHOLAR LDA-COX, for which we provide a complete summary of all topics learned for the seven datasets along with a detailed look at a few datasets. We remark that clinical expertise is required to interpret the topics.

We begin with summaries of the topics learned. Back in Section 1, we already presented one such summary for the SUPPORT3 dataset in Table 1. The summaries for the rest of the datasets are in Tables 5, 6, 7, 8, 9, and 10. For each topic, we state both the Cox β regression coefficient as well as the topic interpretation. For all datasets except MIMIC-ICH, larger β corresponds to *shorter* mean/median survival time. For MIMIC-ICH, larger β corresponds to *shorter* mean/median hospital length of stay. Note that sometimes, spurious topics are found, where a clinical interpretation readily reveals that we could have used a fewer number of topics (although the hyperparameter selection procedure we use that chooses the best model based on validation C^{td} index would not know this). Overall, seeking a clinical interpretation of topics was straightforward. In contrast, when, for example, we presented topics learned using a neural survival-supervised topic model that encouraged sparsity, a clinical expert was unable to determine what the topics meant, with a key problem raised being that the features that are most probable per topic did not appear to be related to each other. We suspect that this has to do with the known issue with lasso regularization where within a group of features that have high pairwise correlation, lasso will arbitrarily choose one of these features and give 0 weight to the others [Zou and Hastie, 2005, Section 2.3].

To obtain the topic interpretations for each dataset, we filter out features that appear in too few or too many patients. Importantly, following the work of Schofield et al. [2017], we filter features *after* learning a topic model in contrast to doing so *before* learning the model. Schofield et al. empirically find no advantage in filtering features before learning a topic model compared to doing it afterward. For our purposes, filtering features before learning a topic model presents problems since there are too many possible ways to do this filtering, and it is unclear how these different filtering approaches impact the topics that are learned. Dawson and Kendziorski [2012] for example use a heuristic preprocessing step in how they use SURVLDA where they cluster subjects based on their survival outcomes and screen out features that are not sufficiently different between the clusters. The problem is that there are far too many choices of how to do this clustering and how to decide what features are sufficiently different even before learning the topic model. By instead filtering features after learning the model, we leave this choice up to the user to specify. The benefits are that there is no need to retrain the model when we try different filters, and moreover, the filtering is fast so it can be adjusted on demand, for example accounting for clinician input. For the results that we show on learned topics by SCHOLAR LDA-COX, we specifically filter out features that appear in fewer than 2% of the patients or more than 50% of the patients. Essentially features that are too rare do not help explain enough of the patient cohort, and features that are too common do not help with interpretation. We tried different thresholds and found ones that appear to work reasonably well across all datasets.

Table 5: Summary of topics learned by SCHOLAR LDA-COX on the SUPPORT1 (acute respiratory failure, multiple organ system failure) dataset. Higher β is associated with shorter survival time.

β	Topic interpretation
0	with cancer, metastases, electrolyte abnormalities, vitals
-5.05	protective, female, diabetic
-5.43	protective, young, no comorbidity

Table 6: Summary of topics learned by SCHOLAR LDA-COX on the SUPPORT2 (COPD, congestive heart failure, cirrhosis) dataset. Higher β is associated with shorter survival time.

β	Topic interpretation
5.30	old, comorbid
2.72	middle age, less comorbid, tachycardia
0	Young healthy baseline, tachycardia

Table 7: Summary of topics learned by SCHOLAR LDA-COX on the SUPPORT4 (coma) dataset. Higher β is associated with shorter survival time.

β	Topic interpretation
0.47	kidney failure, tachycardia, hypertensive, comorbid
0.08	respiratory distress/MV, infection/inflammation, hypothermic
0.01	hypothermic otherwise normal
0	normal baseline
-0.00011	kidney failure, old, infection/inflammation
-0.58	healthy

Table 8: Summary of topics learned by SCHOLAR LDA-COX on the METABRIC (breast cancer) dataset. Higher β is associated with shorter survival time.

β	Topic interpretation
1.29	er- pr- her2+, high mortality, advanced grade
0	similar to 1, focus on group 4 not 1, site 1 not 3
-1.20	protective her2_status1 (-) er- pr-
-1.29	protective but high cellularity luma; pr+ er+
-1.37	these last two topics are both on protective low npi
-1.38	

Table 9: Summary of topics learned by SCHOLAR LDA-COX on the UNOS (heart transplant) dataset. Higher β is associated with shorter survival time.

β	Topic interpretation
6.92	old, old donor, renal failure, with transfusions, liver failure, previous transplant
0	baseline, heart failure, diabetes, with lvad
-1.45	panel reactive antibodies, middle age, low ischemic time, inotropes, body measurements (height weight bmi)
-5.04	pediatric cases, young, donor with infection
-5.09	*these last two topics appear to be spurious and are a mix of the topics*
-5.17	*with β coefficients 0 and -5.04*

Table 10: Summary of topics learned by SCHOLAR LDA-COX on the MIMIC-ICH (intracerebral hemorrhage) dataset. Higher β is associated with shorter hospital length of stay.

β	Topic interpretation
2.08	relatively healthy, anticoagulated, protective demographic factors
1.34	severe anemia, renal failure, inflammatory profile
1.14	hematuria, thrombocytopenia
0	negative drug screening
-2.05	glycosuria screen, electrolyte abnormalities

In addition to filtering features, we also provide heatmap visualizations. These heatmaps were presented to a clinician to obtain the summaries in Tables 1, 5, 6, 7, 8, 9, and 10. In Section 1, we already presented one such heatmap for the SUPPORT3 dataset in Figure 1. Heatmaps for the other datasets are shown in Figures 2, 3, 4, 5, 6, and 7; note that for the UNOS and MIMIC-ICH datasets, due to the large number of features, we truncate the heatmap to only show the top $\sim$80 features (since we only display categorical variables as a block of features at once, we do not get to exactly 80). In these heatmaps, the columns index different topics (with Cox β regression coefficient displayed per topic; the topics are sorted in decreasing order of β coefficient). The rows index different features. The features are sorted based on the maximum word probability across topics (i.e., for the k-by-d topic-word matrix A, for the v-th column/word, we sort based on the score $\max_{g=1,\ldots,k} A_{g,v}$). Furthermore, after doing this sorting, we group together features corresponding to the same categorical variable. Note that we only show features that meet the filtering requirements stated previously.

In producing these heatmaps, we also tried a few variations on the plots to present to a clinician. We sorted the words instead based on the largest difference between word probabilities across topics (i.e., rank words based on the score $(\max_{g=1,\ldots,k} A_{g,v}) - (\min_{g=1,\ldots,k} A_{g,v})$ for the v-th word) and also based on the average probability across topics ($\frac{1}{k}\sum_{g=1}^{k} A_{g,v}$). Qualitatively, we did

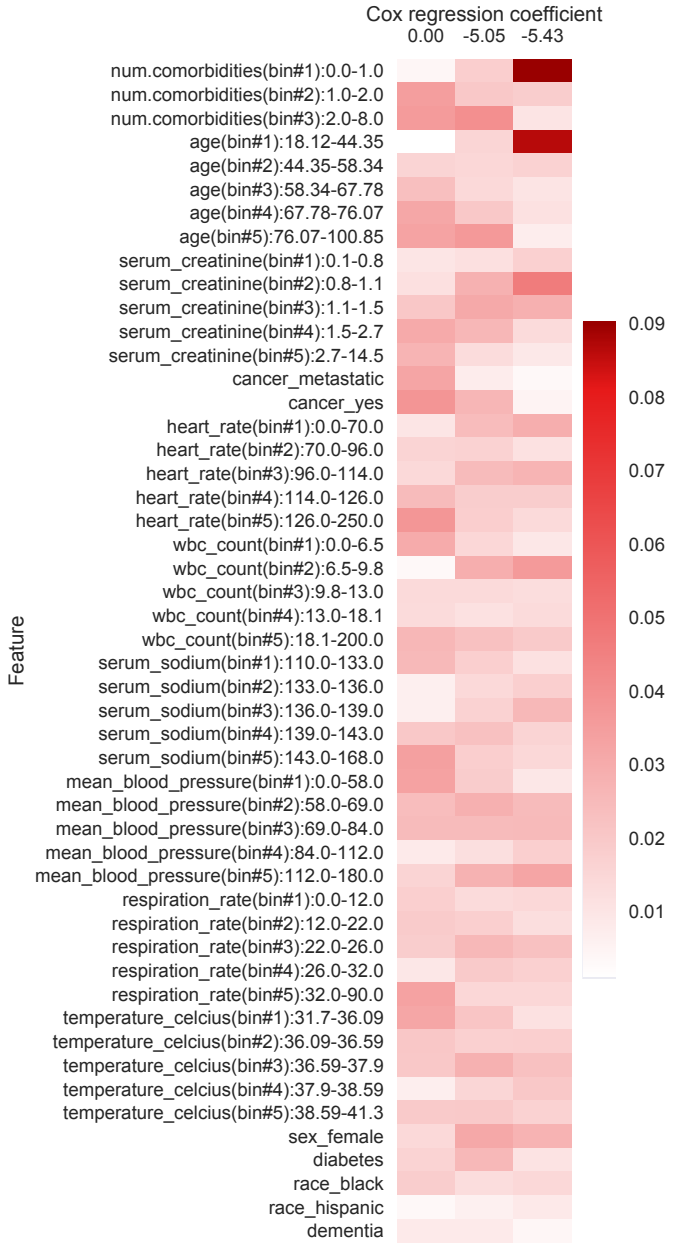


Figure 2: Topics learned by SCHOLAR LDA-COX on the SUPPORT1 (acute respiratory failure/multiple organ system failure) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic.

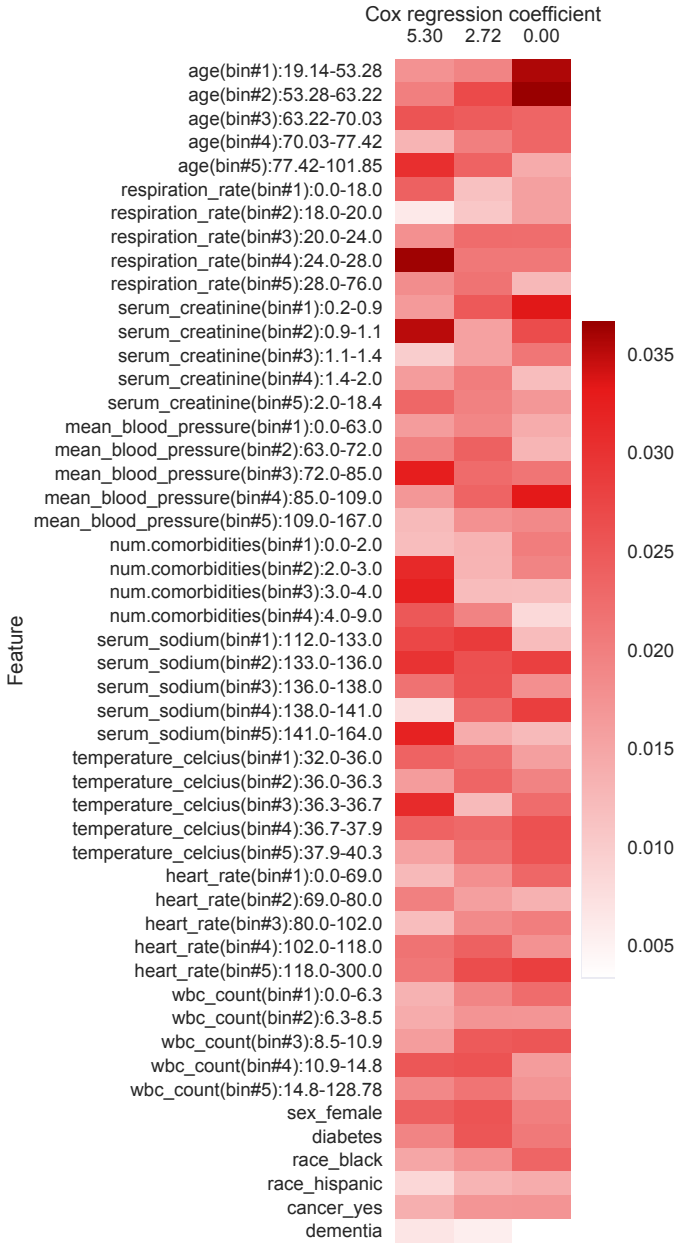


Figure 3: Topics learned by SCHOLAR LDA-COX on the SUPPORT2 (COPD/congestive heart failure/cirrhosis) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic.

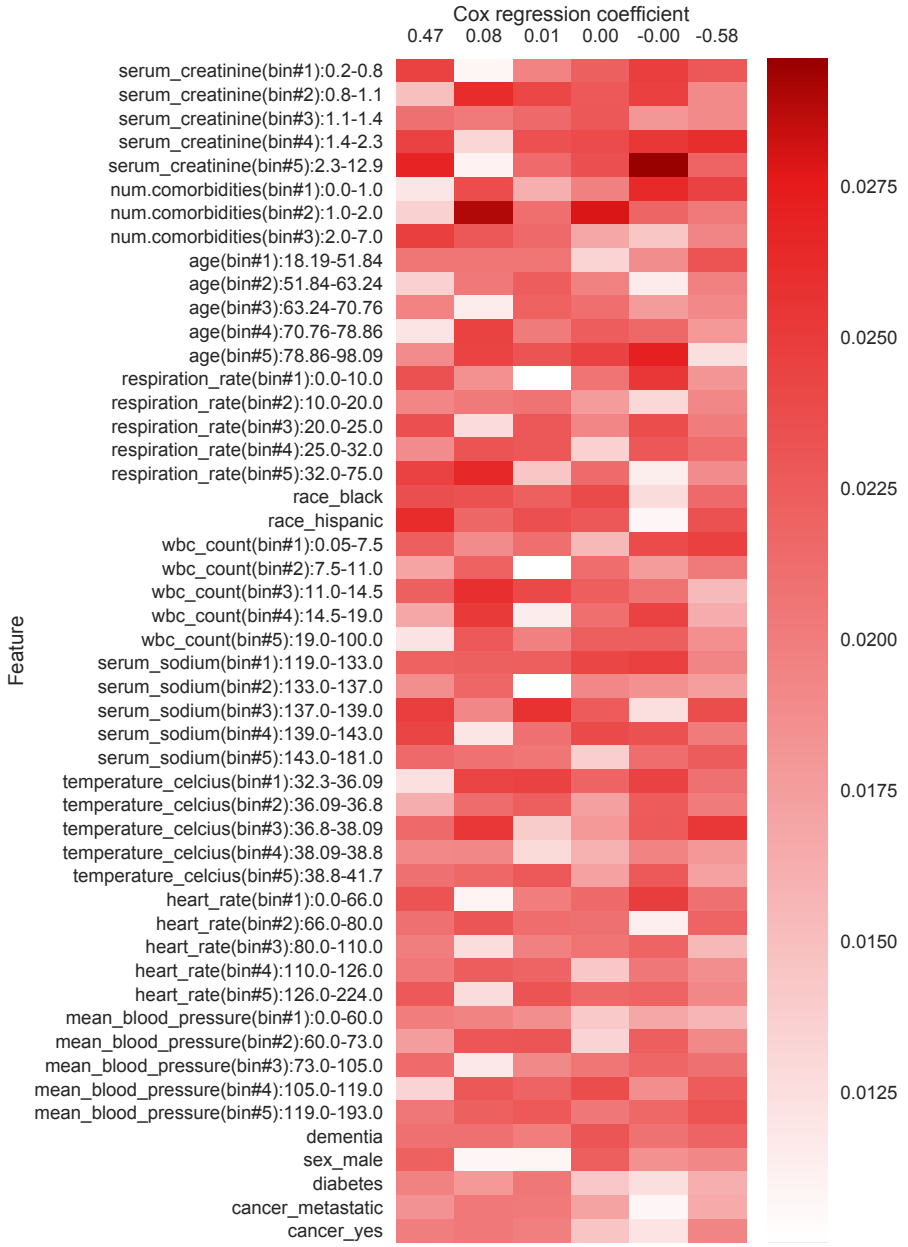


Figure 4: Topics learned by SCHOLAR LDA-COX on the SUPPORT4 (coma) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic. Note that the Cox regression coefficient -0.00 actually corresponds to a value of -0.00011.

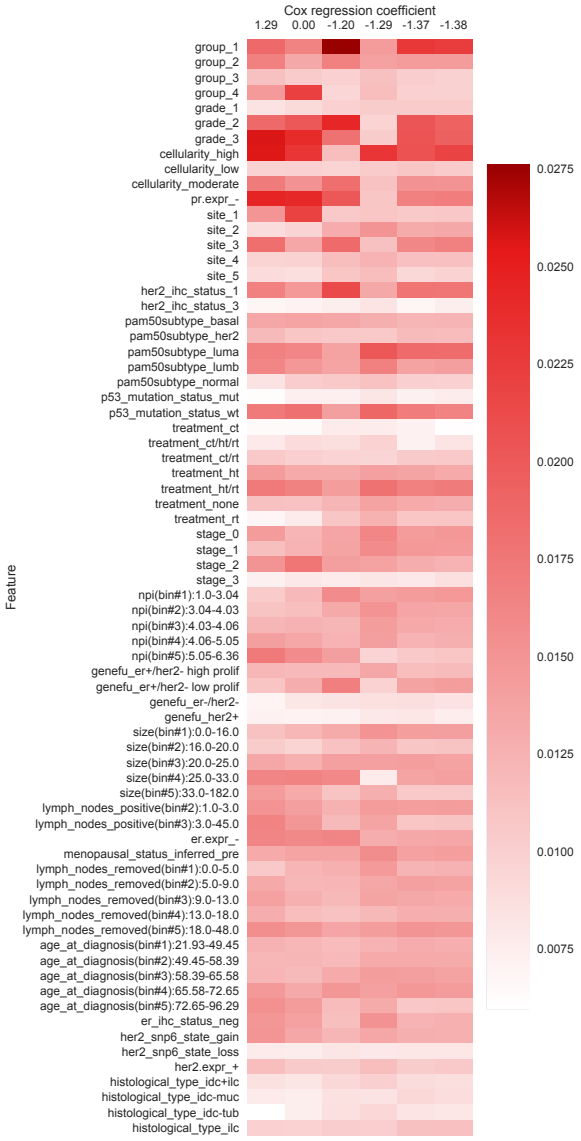


Figure 5: Topics learned by SCHOLAR LDA-COX on the METABRIC (breast cancer) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic.

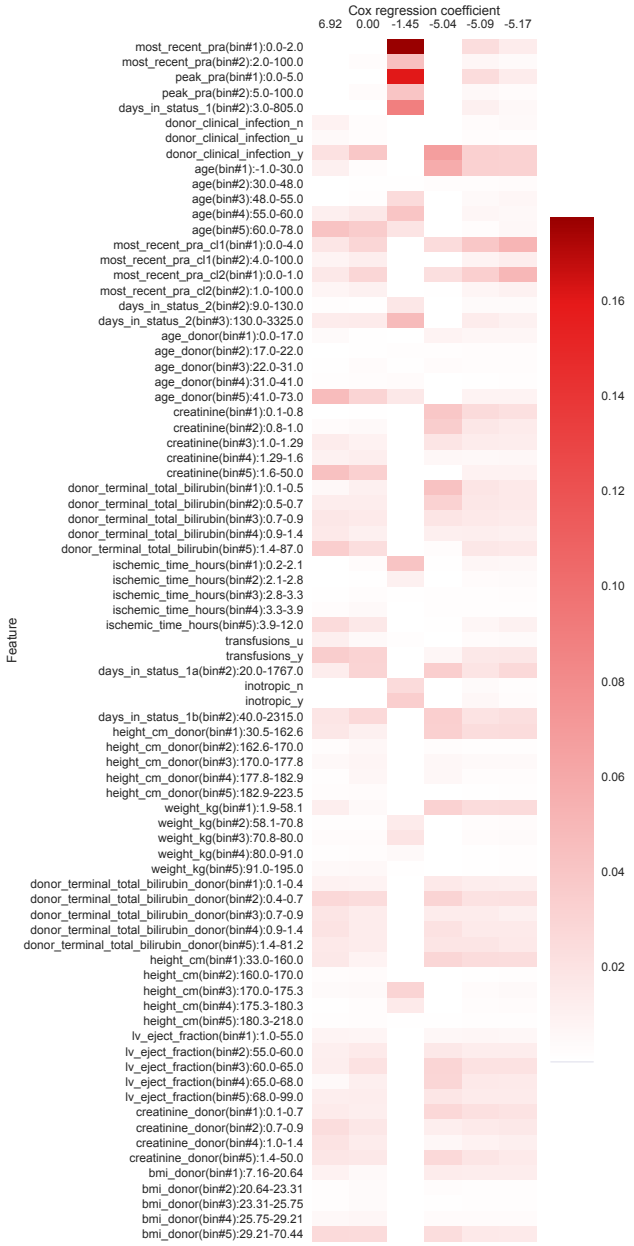


Figure 6: Topics learned by SCHOLAR LDA-COX on the UNOS (heart transplant) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic.

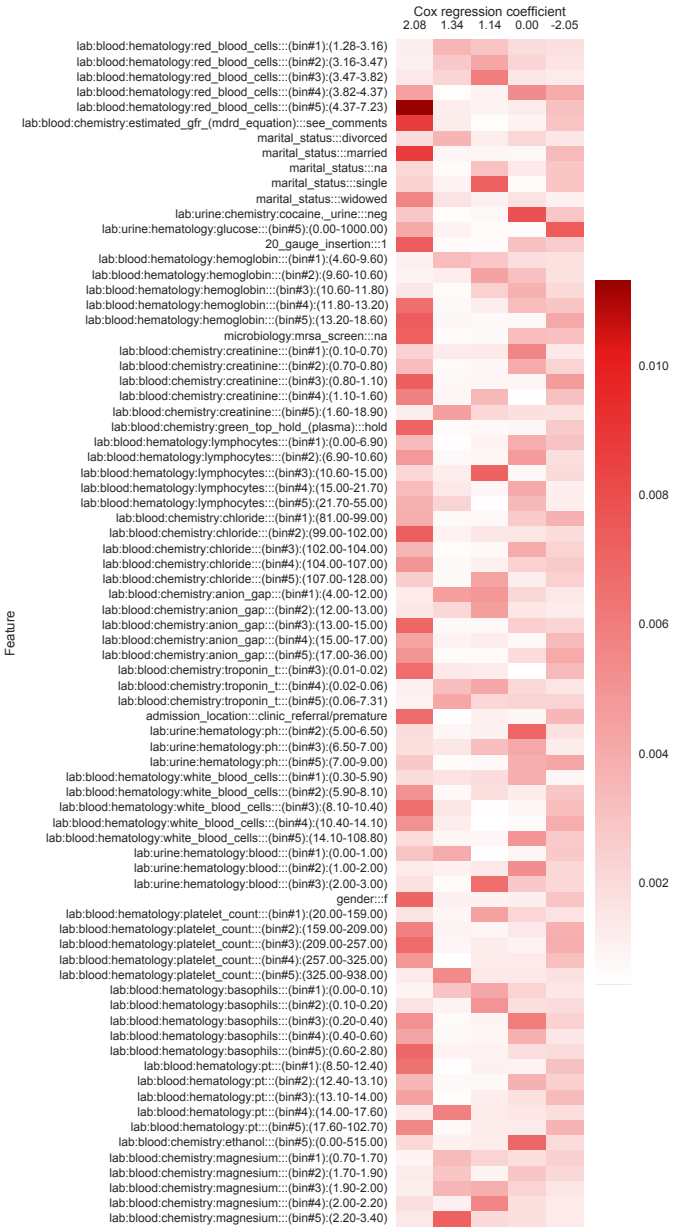


Figure 7: Topics learned by SCHOLAR LDA-COX on the MIMIC-ICH (intracerebral hemorrhage) dataset. Columns index topics and rows index features/"words". The values are probabilities of each feature conditioned on being in a topic.

not find an advantage to using these compared to the score we first presented of using the maximum word probability across topics. We also tried instead of using the raw word probabilities per topic, re-ranking words based on the topic TF-IDF score by Blei and Lafferty [2009, equation (4.3)] and also based on the IDF score by Alokaili et al. [2019]. Qualitatively, we found that the topic TF-IDF weighting highlights a few words per topic but this weighting can be a bit too aggressive (the few words highlighted could be hard to interpret). IDF weighting could help draw out underrepresented words. Overall, though we did not see a clear advantage to using TF-IDF or IDF weighting in presenting the heatmap visualizations.

Note that prior to using our heatmap visualizations, we first tried providing a clinician with a listing of most probable words per topic. This is a standard approach for interpreting LDA models for text data. However, this way of conveying information turned out to be difficult for the clinician to quickly parse. For example, a feature might be in the top 20 most probable words for two different topics, and at that point understanding the difference in how probable the feature is across the two topics would be helpful. A listing of top words per topic did not make it easy to quickly find this information. For this reason, we switched to a heatmap visualization where each row of the heatmap directly gives us a quick way to compare probabilities of a feature/word across topics.

5 Discussion

Despite many methodological advances in survival analysis with the help of deep learning, these advances have predominantly not focused on interpretability. Model interpretation can be especially challenging when there are many features and how they relate is unknown. In this paper, we show that neural survival-supervised topic models provide a promising avenue for learning structure over features in terms of "topics" that help predict time-to-event outcomes. These topics can be used by practitioners to check if learned topics agree with domain knowledge and, if not, to help with model debugging.

Our work thus far has a number of limitations. We discuss some of these limitations next.

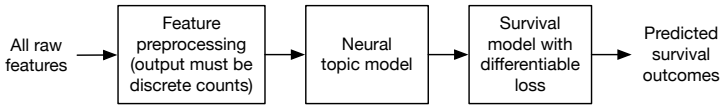
Moving beyond discrete data Our focus has been on when the raw features are encoded in a format specifying whether different historical clinically relevant events occur or not (the "words" of the topic model). This encoding inherently is discrete. The discretized raw counts then get modeled by a neural topic model, and the topics are treated as the input "features" for the survival model, as shown in Figure 8(a). Discretizing continuous data inherently results in some loss in information. Better understanding how different discretization strategies (such as those described in A.3) impacts learned neural survival-supervised topic models in terms of accuracy and interpretability is an important direction for future research. Note that it is possible to also have

some user-specified raw features be modeled directed by the survival model rather than being modeled by the topic model first, as shown in Figure 8(b); in this case, the raw features directly modeled by the survival model need not be discretized. For example, depending on the problem, we may want to have age be directly modeled by the survival model (e.g., a Cox model) rather than being explained by the topic model. As another example, consider gender being directly modeled by the survival model and not provided to the topic model. We could still try to understand how gender relates to the topics learned by adding interaction terms for the survival model (e.g., an indicator variable specifying whether female and topic 1 jointly occurs, whether female and topic 2 jointly occurs, etc).

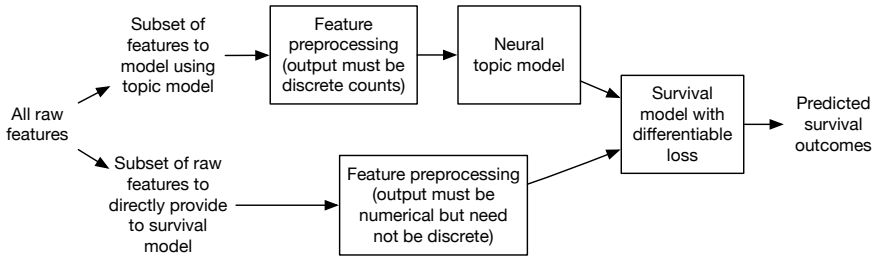
Separately, much of the same ideas we presented in interpreting neural topic models readily apply to *prototypical part networks* (ProtoPNets) [Chen et al., 2019, Ming et al., 2019], which behave like neural topic models but for raw data that are images or time series. Note that ProtoPNets can directly work with continuous-valued features without discretization. For example, given an input image, a ProtoPNet transforms the image into a vector representation specifying how much of each of k different prototypes are present in the image ("similarity scores" that are nonnegative); this vector representation behaves much like the topic weight vectors W_i's that we have considered and could be fed as input to a survival model incorporating a background topic. Using these ideas, it is possible to build survival-supervised neural topic models that accept heterogeneous inputs, for example using the discrete "words" that we have considered in this paper, alongside images and time series (that could be left as continuous-valued). Of course, we could again choose some features to be directly modeled by the survival model. The overall diagram depicting this setup is shown in Figure 8(c).

Incorporating additional structure in topics Topics learned by neural survival-supervised topic models vary in how easy they are for a clinician to interpret. We suspect that to improve interpretability, additional regularization is essential. For example, one possible research direction is to automatically find clinical measurements that do not plausibly co-occur within individual subjects, and add regularization that disallows these "contradictory" clinical measurements from both being highly probable within the same topic. For example, hematocrit and hemoglobin should be highly correlated, so we would expect that if a topic says one has a high probability of taking on a high value, then the topic should also say that the other has high probability of taking on a high value.

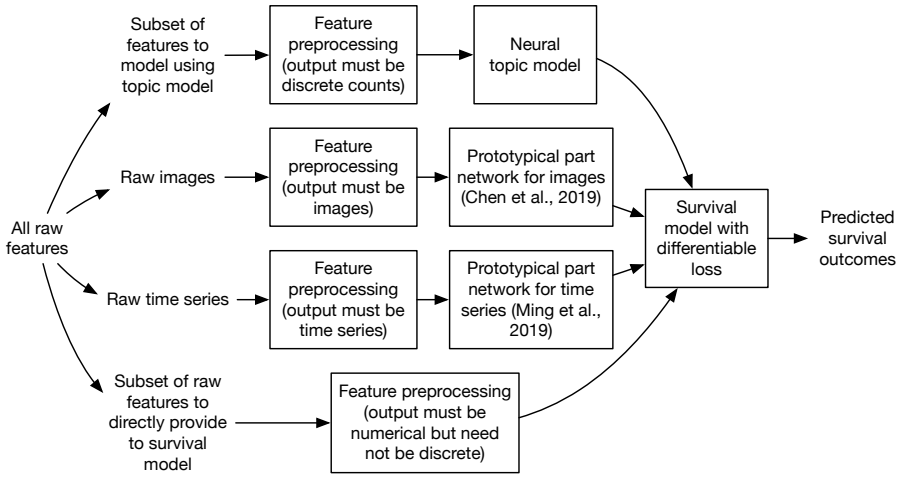
As another example, when a continuous measurement is discretized, we currently do not impose any constraints on the resulting discretized variables even though they are, of course, highly dependent on each other (i.e., a continuous variable is converted into a collection of variables that correspond to whether different discretization bins occur, and when one of them occurs, we know that the others cannot occur). A fix to this issue would be to add in loss



(a)



(b)



(c)

Figure 8: Incorporating different raw feature types: (a) our framework, (b) an extension of our framework allowing some raw features (which need not be discretized) to be directly modeled by the survival model, and (c) an extension of our framework that also uses prototypical part networks [Chen et al., 2019, Ming et al., 2019] that are in some sense like topic models but for images and time series (we can omit different parts of this general framework depending on the raw input data that are available, e.g., if images are not available, then we remove the part involving prototypical part networks for images).

terms to say when specific "words" explicitly do *not* occur.

A less straightforward relationship to encourage is that a specific continuous variable (that has been discretized) should have a monotonic association with the survival time. Neither the raw continuous variable nor its corresponding discretized variables are provided directly as input to the survival model—instead they are treated as inputs to the topic model. One possible workaround is as follows. Suppose that we think age should have a monotonic association with survival time, and that it is discretized into bins 1 through 5, going from smaller to larger ages. Then for a specific topic, we could constrain the topic's probabilities for the discretized variables for age to be monotonic (i.e., the probabilities of the bins either increase from bin 1 up to bin 5, or they decrease from bin 1 up to bin 5 depending on whether we want the presence of the topic to be associated with higher or lower ages).

Topic stability As a separate direction that requires further investigation, thus far, we have not conducted experiments to quantify how "stable" the topics learned are across, for example, different random neural net parameter initializations. This is a problem more broadly found in training neural networks and is referred to as "prediction churn" [Bahri and Jiang, 2021]. Better understanding how much the learned topics change due to random initialization would be helpful. We suspect that introducing regularization—such as the one we suggested for encouraging plausible co-occurrences—would lead to more stable topics learned. Even if we develop an improved understanding of topic stability, we would further need to understand how best to communicate this information to clinicians.

Competing risks In this paper, we focused on the standard right-censored survival analysis setup. We point out that our framework trivially extends to the competing risks setting, where we further want to reason about the cause of death (or more generally, a collection of competing critical events that could occur, where whichever occurs first prevents the other critical events from occurring). In this case, for each training subject, we assume that in addition to the subject's raw clinical events data, observed time, and indicator variable for whether death occurred, if death did occur, we also know the cause of death (among a finite set of causes under consideration). Standard competing risk models (e.g., see Chapter 8 of the textbook by Kalbfleisch and Prentice [2002]) can be used in place of the survival model in our neural net framework to obtain a neural topic model for competing risks. For example, one approach would be to have a Cox loss per cause of death, where the key idea here is that standard competing risk models still can be framed as minimizing a differentiable loss function (specifically a negative log likelihood). Empirically studying the resulting neural topic models for competing risks could provide interesting practical insights, with the goal of automatically surfacing feature relationships through a topic model, and finding associations between topics and the different causes of death.

Theoretical analysis Lastly, we mention that developing theory to understand when and why neural survival-supervised topic models work would be valuable. In particular, for what datasets should we expect to be able to learn such neural topic models that have sufficiently high prediction accuracy and are also easy to interpret? What special structure should be present in the data and how much data do we need? How does data preprocessing (e.g., discretization) impact these neural topic models? Finding theory that answers these questions could help clinicians understand when our proposed framework is most effective and what the best practices are in collecting and preprocessing data for use with our framework.

A Datasets and Preprocessing Details

We describe the seven datasets we use and how we preprocess the data to obtain feature vectors of the format specified in Section 2.

A.1 Datasets

SUPPORT The dataset from the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) [Knaus et al., 1995] is freely available online.[6] This dataset contains 14 clinical features collected from seriously ill hospitalized adults, such as their age, presence of cancer, and neurologic function. These features were collected from patients on the third day after the study started, and patients were followed for survival until 180 days after the study entry. For our purposes, the dataset was split into four datasets corresponding to different disease groups (acute respiratory failure/multiple organ system failure, cancer, coma, COPD/congestive heart failure/cirrhosis), as done by Harrell [2015]. After we created these four subsets, all subjects from the cancer group have identical values for a clinical feature related to cancer presence, so this feature was removed only for the cancer cohort, resulting in 13 clinical features for the SUPPORT3 dataset.

METABRIC The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset is available on the Synapse platform[7]. This dataset contains clinical and genetic features from breast cancer patients, and their respective survival durations. We only used a subset of 24 features that are available for open use through Synapse. This dataset includes 1981 breast cancer patients in total, around 55.2% of whom were censored and not followed until death. The original METABRIC paper [Curtis et al., 2012] discusses how the dataset's clinical features were defined in more detail.

[6] http://biostat.mc.vanderbilt.edu/wiki/Main/SupportDesc
[7] https://www.synapse.org/

UNOS The UNOS dataset was extracted from the United Network for Organ Sharing (UNOS) database[8], and curated in order to replicate the pre-processing documented by Lee et al. [2018] and Yoon et al. [2018]. We selected only patients who went through heart transplantations in the 30-year window from January 1985 to December 2015. Because Yoon et al. [2018] did not document the exact list of feature names that we could directly extract from the database, we attempted to the best of our ability to curate a list of features that overlaps the most with the feature table presented by them. We ended up with 49 features in total, among which 31 are recipient-related, 12 are donor-related, 6 are compatibility related. For this dataset, our objective is to predict patients' post-transplantation survival time. Because we assumed December 2015 to be the end of data collection, patients who were still alive as of December 2015 are all considered censored samples. Among 62644 patients who underwent transplantation, around 50.2% are censored samples.

MIMIC-ICH The intracerebral hemorrhage (ICH) dataset we evaluated on is created from MIMIC-III (version 1.4), a critical care health records database containing 52 thousand individuals and their hospital encounters involving admission to the ICU at Beth Israel Deaconess center between 2001 and 2012 [Johnson et al., 2016]. Experiments were conducted using a subset of the MIMIC-III data consisting of patients having spontaneous intracerebral hemorrhage requiring admission to the ICU. Patients were included in the study if they have an ICU admission with a primary billing code of intracerebral hemorrhage, resulting in a cohort of 961 individuals. For patients who are admitted to the ICU multiple times, we only consider their first visit to the ICU within the dataset. We aim to predict patients' lengths of stay in the ICU (specifically time until discharge). This subset of the data has no right-censoring in the sense of data no longer being collected midway through a patient's ICU stay. However, 23.1% of the patients die in the ICU; for these individuals, we record the time until death as the observed time and set the indicator variable for whether the patient is discharged to 0. In particular, death is effectively treated as the sole censoring event.

Features extracted include demographics, medications, billing codes, procedures, laboratory measurements, events recorded into charts, and vitals. Features were extracted from the relational database into a 4-column format for *patient id*, *time*, *event*, and *event value*. To prevent erroneous merging of different events into a single event, and to provide more informative events, event strings are concatenations of the event descriptor prefixed with the table from which they are derived and additional relevant information such as measurement type, measurement units, etc. Because events recorded in charts are sometimes automated and sometimes manually entered, a physician-developed mapping and lower-casing all fields were used to resolve duplicate entries. As we aim to predict the patient length of stay in ICU, we extract clinical events from the subjects' electronic health records strictly before ICU admission. After

[8] `https://www.unos.org/data/`

preprocessing, the total number of features used for prediction is 1530.

A.2 Features Used

For all of our datasets, categorical features were one-hot encoded. Specifically to the Cox proportional hazards and lasso-regularized Cox baselines, for each categorical feature, one category was removed as the reference column. For methods that use topic modeling, we realized it does not make sense to encode numeric clinical events as they are. Instead, numeric clinical events were treated as categorical by mapping observed values to equally spaced ranges by quintile (5 bins of roughly equal number of subjects per bin). When values of a numeric clinical event are highly cluttered (i.e., the 20/40/60/80 percentile values of the event do not correspond to 4 unique threshold values so that there end up being fewer than 5 bins), we allow the number of bins to be less than 5, where the resulting bins can have imbalanced numbers of subjects. For instance, if there are fewer than 5 unique values for the clinical event across data points, then we cannot discretize the event into 5 nonempty bins.

Features for the MIMIC-ICH dataset were created slightly differently. Our definition of clinical events mean that a subject can have multiple instances of one event; for example, one patient might have multiple results for a particular lab test on file. Under this case, single-occurrence categorical events (e.g., gender) were one-hot encoded as usual; multiple-occurrence categorical events (e.g., urine color) were encoded by counting each category's occurrences in a single subject's records. For numeric clinical events, as a subject may have a list of numeric values recorded, we engineered numeric features that captured the minimum, maximum, median, and length of a subject's list of recordings. However, this was not necessary for methods that use topic modeling, because mapping values to equally spaced bins took care of multiple-occurrence numeric events for us.

We would also like to note that missing records were not imputed as missing certain events can have clinical significance. Therefore, for features with incomplete records, the missing entries were first filled with zeros, and then an additional feature was added solely to indicate whether missingness is observed for each subject; this approach to handling missing data is motivated by the work of Lipton et al. [2016]. While we added features that solely indicate missingness for all baseline methods, methods that use topic modeling do not require encoding missingness explicitly. For topic modeling based methods, feature vectors encode number of occurrences, so a patient with missing feature simply has that feature's number of occurrences set to 0. For this reason, we did not explicitly encode missingness as a separate feature for methods that use topic modeling.

A.3 Other Possible Ways to Encode Clinical Measurements

Our feature preprocessing has largely been chosen to be relatively easy to explain. We now mention other strategies that are possible for discretization and,

separately, for summarizing a feature across time.

Discretization We discretize continuous features into quintiles (as we mentioned earlier, sometimes this is not possible so we simply use fewer than 5 bins), which is a simple strategy that can be used for different continuous features without a priori knowledge. However, if one did have domain knowledge about how specific features could be discretized, then such discretization strategies could be used instead of the simple quintile binning strategy. As an example, there are specific cutoffs whereby cohorts are defined (e.g., lactate levels of 4), and where medical interventions are indicated (e.g., mean arterial pressures below 65).

Alternatively, one could even learn how to discretize a specific continuous feature (a single real number). For instance, taking the feature's value across the training data, we could use a user-specified clustering algorithm (e.g., Jenks natural breaks [Jenks, 1967]) to cluster on the observed values of the continuous feature to decide on how to discretize (the thresholds could come from the boundary points between clusters). A different strategy is to learn a decision tree for survival analysis using a single continuous feature across the data. Such a tree could be learned greedily (using the same tree learning strategy as in random survival forests [Ishwaran et al., 2008]) or optimally by solving a mixed-integer program [Bertsimas et al., 2020]: the leaves of the learned tree directly correspond to the discretization bins. A generalization of this idea is possible in which multiple continuous features could be discretized together (train a single decision tree with these different features and then let the final tree leaves correspond to the discretization bins).

Summarizing a feature across time For ease of exposition, we had simply counted how often a feature occurred across time to obtain the raw counts matrix X. If we had domain knowledge of how a specific feature should be summarized across time, then we could take this into account when summarizing the feature. For example, if we take many oxygen saturation measurements within a few minutes, clinically we typically take the highest measured value because the physiologic process prevents rapid fluctuations in saturation, and the measurement is intended to grossly assess oxygenation and perfusion. Alternatively, we could use the approach by Johnson et al. [2021] that automatically learns how to summarize continuous or discrete features across time in such a way that the summary features are clinically interpretable. Each summary feature can then be discretized using any user-specified discretization strategy, such as the clustering or decision-tree approaches we described in the previous paragraph.

Table 11: Hyperparameter grids used during model training.

Model	Hyperparameter Grid
COX	lasso regularization weight: 0, 0.0001, 0.001, 0.01, 0.1, 1.0
RSF	number of trees: 100 number of features used per split: sqrt of total number of features, rounded up max depth: 2, 4, 6, 8
DEEPSURV	number of hidden layers for the multilayer perceptron: 1, 2, 4 number of nodes per hidden layer: 16, 32, 64
DEEPHIT	number of hidden layers for the multilayer perceptron: 1, 2, 4 number of nodes per hidden layer: 16, 32, 64 number of durations (in time discretization): 64, 128 α (in original DeepHit paper; not LDA Dirichlet hyperparameter): 0.1, 0.5, 0.9 σ (in original DeepHit paper; not AFT scale parameter): 0.1, 1.0, 10.0
NAIVE LDA-COX	number of topics: 2, 3, 4, 5, 6
SCHOLAR LDA-COX	number of topics: 2, 3, 4, 5, 6 word embedding dimension: 16 32, 64 $\lambda_{\text{survival}}$: 1, 100, 10000, 1000000
SCHOLAR LDA-AFT	number of topics: 2, 3, 4, 5, 6 word embedding dimension: 16 32, 64 $\lambda_{\text{survival}}$: 1, 100, 10000, 1000000 λ_{ranking}: 1
SCHOLAR SAGE-COX	number of topics: 2, 3, 4, 5, 6 word embedding dimension: 16 32, 64 $\lambda_{\text{survival}}$: 1, 100, 10000, 1000000 $\lambda_{\text{small-deviation}}$: 0.005, 0.05, 0.5, 5
SCHOLAR SAGE-AFT	number of topics: 2, 3, 4, 5, 6 word embedding dimension: 16 32, 64 $\lambda_{\text{survival}}$: 1, 100, 10000, 1000000 λ_{ranking}: 1 $\lambda_{\text{small-deviation}}$: 0.005, 0.05, 0.5, 5

B Hyperparameter Search

We use grid search, with the same grid of hyperparameters used across datasets per model as given in Table 11. For neural net approaches, we always train using Adam [Kingma and Ba, 2014] with a batch size of 256 and use early stopping (no improvement in best validation C^{td} index within 10 epochs) with a budget of 512 epochs; however we do vary the learning rate and sweep over the choices of 0.01 and 0.001.

Acknowledgments

This work was supported in part by Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commerical products, or organizations imply endorsement by the U.S. Government.

References

Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. Re-ranking words to improve interpretability of automatically generated topics. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 43–54, 2019.

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

Dara Bahri and Heinrich Jiang. Locally adaptive label smoothing for predictive churn. *arXiv preprint arXiv:2102.05140*, 2021.

Dimitris Bertsimas, Jack Dunn, Emma Gibson, and Agni Orfanoudaki. Optimal survival trees. *arXiv preprint arXiv:2012.04284*, 2020.

David M. Blei and John D. Lafferty. Topic models. *Text Mining: Classification, Clustering, and Applications*, 10(71):34, 2009.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Norman Breslow. Discussion of the paper by D. R. Cox (1972) cited below. *Journal of the Royal Statistical Society, Series B*, 34(2):216–217, 1972.

Dallas Card, Chenhao Tan, and Noah A. Smith. Neural models for documents with metadata. In *Proceedings of Association for Computational Linguistics*, 2018.

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.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32:8930–8941, 2019.

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

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

John A. Dawson and Christina Kendziorski. Survival-supervised latent dirichlet allocation models for genomic analysis of time-to-event outcomes. *arXiv preprint arXiv:1202.5999*, 2012.

Adji B. Dieng, Francisco J.R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8: 439–453, 2020.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. Sparse additive generative models of text. In *International Conference on Machine Learning*, pages 1041–1048, 2011.

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

Frank E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer, 2015.

Frank E. Harrell, 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.

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

George F. Jenks. The data model concept in statistical mapping. *International Yearbook of Cartography*, 7:186–190, 1967.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, 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, 2016.

N. Johnson, S. Parbhoo, A. Ross, and F. Doshi-Velez. Learning predictive and interpretable timeseries summaries from ICU data. In *Proceeding at the Conference on American Medical Informatics Association (AMIA)*, volume 1, pages 1–10, 2021.

John D. Kalbfleisch and Ross L. Prentice. *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, 2nd ed. edition, 2002.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

John P. Klein and Melvin L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated data*. Springer Science & Business Media, 2006.

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

Håvard Kvamme and Ørnulf Borgan. Continuous and discrete-time survival prediction with neural networks. *Lifetime Data Analysis*, 27(4):710–736, 2021.

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.

John D. Lafferty and David M. Blei. Correlated topic models. In *Advances in Neural Information Processing Systems*, pages 147–154, 2006.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *AAAI Conference on Artificial Intelligence*, 2018.

Zachary C Lipton, David C Kale, Randall Wetzel, et al. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare*, 56, 2016.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

Jon D. McAuliffe and David M. Blei. Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128, 2008.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2019.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577, 2008.

Ross L. Prentice. Linear rank tests with right censored data. *Biometrika*, 65(1): 167–179, 1978.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Alexandra Schofield, Måns Magnusson, and David Mimno. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 432–436, 2017.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 2011.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*, 2017.

Harald Steck, Balaji Krishnapuram, Cary Dehing-oberije, Philippe Lambin, and Vikas C Raykar. On ranking in survival analysis: Bounds on the concordance index. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Jinsung Yoon, William R. Zame, Amitava Banerjee, Martin Cadeiras, Ahmed M. Alaa, and Mihaela van der Schaar. Personalized survival predictions via trees of predictors: An application to cardiac transplantation. *PloS One*, 13(3), 2018.

He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. Topic modelling meets deep neural networks: A survey. *arXiv preprint arXiv:2103.00498*, 2021.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.