Survival Analysis of High-dimensional Data with Graph Convolutional Networks and Geometric Graphs

Yurong Ling, Zijing Liu, Jing-Hao Xue, *Senior Member, IEEE*

***Abstract*—This paper proposes a survival model based on graph convolutional networks with geometric graphs directly constructed from high-dimensional features. First, we clarify that the graphs used in graph convolutional networks play an important role in processing the relational information of samples, and the graphs that align well with the underlying data structure could be beneficial for survival analysis. Second, we show that sparse geometric graphs derived from high-dimensional data are more favourable compared with dense graphs, when used in graph convolutional networks for survival analysis. Third, from this insight, we propose a model for survival analysis based on graph convolutional networks. By using multiple sparse geometric graphs and a proposed sequential forward floating selection algorithm, the new model is able to simultaneously perform survival analysis and unveil the local neighbourhoods of samples. The experimental results on real-world datasets show that the proposed survival analysis approach based on graph convolutional networks outperforms a variety of existing methods and indicate that geometric graphs can aid survival analysis of high-dimensional data.**

***Index Terms*—Survival analysis, graph convolutional networks, geometric graphs, sequential selection**

I. Introduction

APPLICATIONS of survival analysis can be found in various areas, such as clinical research [1], [2], credit scoring [3] and sociology [4]. The primary objective of survival analysis is to predict the time until the occurrence of the particular event of interest [5], which is also called time-to-event prediction. A major challenge when dealing with time-to-event data is the existence of censored instances, where the event times are unknown and the information of such instances is available only up to specific time points. Thus, it is impractical to directly apply those regression models designed for data without censoring to the task of time-to-event prediction.

To perform survival analysis on data containing censored instances, the Cox proportional hazards model and its penalized extensions are commonly used [6]–[8]. These models assume the proportional hazards and adopt the partial likelihood, possibly with regularisation, to estimate the model coefficients for calculating the relative risks and survival functions of patients. Apart from the Cox model and its variants, numerous machine learning approaches have been proposed for analysing time-to-event data. Random survival forests (RSF), extending random forests to censored data, are introduced with the log-rank test as the splitting rule for growing survival trees [9]. Support vector methods have been successfully adapted for survival analysis by reformulating the survival problem as a regression or ranking problem [10]–[12]. With the aid of multi-task learning, survival models can be improved by efficiently using shared-knowledge of related survival problems [13], [14].

Recently, methods based on neural networks have been applied to time-to-event data due to their ability to learn complex data structures. A common approach is to integrate the Cox regression into the framework of neural networks, by employing the negative Cox log partial likelihood as the loss function. The partial likelihood is a ranking loss that accounts for the ordering of patients' relative risks, and thus using it as a loss function often results in better concordance of survival predictions. Examples of such approaches include DeepSurv and Cox-nnet, which use the multilayer perceptron (MLP) [2], [15]. Alternatively, a number of approaches are proposed to circumvent the constraint of the proportional hazards. The partial likelihood is extended by letting the relative risk function depend on time [16]. In [17]–[19], the authors cast the time-to-event formulation to a discretised-time classification problem and directly model the survival or hazard function at discrete intervals with neural networks. Most aforementioned works based on neural networks aim at improving the ordering of the patients' survival predictions, while the non-proportional extension of the Cox model [16] shows better calibration of survival time estimates. In spite of these successes of neural networks in survival analysis, most studies mentioned above conduct experiments on datasets with the number of features much less than that of samples ($p \ll n$). It is therefore in doubt that their performances on datasets of high dimensionality ($p \gg n$) are still promising.

Graph convolutional networks (GCNs), where the convolution is extended from grid-like data to graph-like ones, have witnessed successes in many classification tasks [20]–[22]. Under the framework of GCNs, one can exploit both sample features and their relational information, which are represented in the form of graphs. Compared with the methods that do not consider a graph and solely utilise sample features, GCNs require the alignment between the graph and the class labels of data as the essence for achieving better classification performance [23]. Concretely, the graph convolution operation in GCNs aggregates and exchanges information between

Y. Ling and J.-H. Xue are with the Department of Statistical Science, University College London, U.K. (e-mail: yurong.ling.16@ucl.ac.uk).

Z. Liu is with the Department of Mathematics, Imperial College London, U.K.

(Corresponding author: Yurong Ling)

neighbouring nodes (samples) in the graph and it is equivalent to a special form of Laplacian smoothing; that is, GCNs make the features of neighbours more similar [24]. Thus, if a graph aligns well with class labels in the sense that neighbouring nodes tend to belong in the same class, a GCN with the graph would improve the class separation by making the features of the samples within a class more homogeneous and those between different classes more heterogeneous. On the contrary, with a graph that is inconsistent with class labels, the features between different classes would be homogenised by the GCN, in which case the GCN is expected to perform worse than the methods considering only sample features. Although people often use available relational information represented in the form of graphs when implementing GCNs, it has been shown that using sparse geometric graphs of samples derived from features themselves in GCNs can improve classification by enhancing the alignment between data and class labels [25].

Inspired by the successful attempts of using GCNs and geometric graphs in the context of classification, we propose a survival model based on GCNs with feature-derived graphs. The above insights into GCNs suggest that the prerequisite of using GCNs to improve survival analysis is that their input graphs align well with sample survival times. That is, linked samples in the graphs should tend to have similar survival times. Otherwise, the graph convolution in the GCNs is inclined to homogenise the samples with markedly dissimilar survival times, and thus adversely affects survival analysis. In this work, we therefore investigate the alignment between geometric graphs derived from features and sample survival times. We also study how to construct graphs that can be used in GCNs to improve survival analysis. In addition, we employ survival data containing large number of features for comparison in this work ($p \gg n$) to study whether GCNs are able to reduce the overfitting, compared with the other approaches based on neural networks. The technical novelty and contributions of our work can be summarised as follows.

First, we show that, for survival analysis, sparse geometric graphs built from high-dimensional features are more favourable for GCNs compared with dense graphs (Sec. III-A). Second, considering that using a single sparse graph may reveal only a small part of the neighbourhood of each sample, particularly for those from high-density areas, we propose to use multiple sparse graphs to uncover the local neighbourhoods (Sec. III-B). Third, we propose a survival model that not only outputs the survival predictions but also captures the local neighbourhoods, by using multiple sparse graphs with GCNs (Sec. III-C). To this end, we first construct various sparse graphs derived from random subsets of features, and each constructed graph is then fed into a GCN with the widely used partial likelihood as the loss function to get the survival forecast. With the principle that the aggregated survival predictions from a set of GCNs would be superior if the union of edge sets of the corresponding graphs approximates better the ground-truth survival time, we select graphs and aggregate their output predictions simultaneously by a proposed sequential forward floating selection algorithm. The experimental results on real-world datasets in Sec. IV demonstrate that the proposed survival model based on GCNs and multiple sparse graphs outperforms state-of-the-art methods. The promising results of the proposed model suggest that geometric graphs could be beneficial for the survival analysis of high-dimensional data.

II. Preliminary Knowledge

In this section, we briefly review some key concepts in survival analysis, the Cox model, GCNs, geometric graphs, and two evaluation metrics for survival models.

A. Survival analysis

Let $T^\star$ denote the time when a particular event of interest occurs. The primary objective of survival analysis is to model the distribution of $T^\star$. Specifically, it can be formulated as the lifetime distribution function

$$F(t) = P(T^\star \leq t) = \int_o^t f(s)ds,$$

where $f(t)$ is the event density function. Equivalently, one can model the survival function defined as the complement of the lifetime distribution function $S(t) = 1 - F(t) = P(T^\star > t)$.

In some survival models like the Cox model, instead of the survival function, the hazard function is learned. The hazard rate function, denoted by $h(\cdot)$, is defined as the event rate at time t conditional on survival until time t or later (that is, $T^\star \geq t$):

$$h(t) = \lim_{dt \to 0} \frac{P(t \leq T^\star < t + dt \mid T^\star \geq t)}{dt} = \frac{f(t)}{S(t)}.$$

The survival function can alternatively be represented in terms of the cumulative hazard function, denoted by $H(\cdot)$:

$$S(t) = \exp[-H(t)],$$

where $H(t) = \int_0^t h(s)ds$.

As we mentioned before, it is possible that the events of interest are not observed for some instances, a situation called censoring. This may occur when we lose track of an individual or the maximum follow-up time is shorter than the survival time. Censoring falls into three groups: right-censoring, left-censoring, and interval censoring. In this paper, we consider the most common right-censoring where the observed survival time is less than or equal to the true survival time. In such scenarios, we observe a possibly right-censored time $T = \min\{T^\star, C\}$, where C is the censoring time.

The feature vector and the observed time for the i-th individual are denoted by $\mathbf{x}_i$ and T_i, respectively. The full likelihood accounting for both censored and uncensored instances is

$$L = \prod_i f(T_i \mid \mathbf{x}_i)^{D_i} S(T_i \mid \mathbf{x}_i)^{1-D_i}, \tag{1}$$

where $D_i = \mathbb{1}_{\{T_i = T_i^\star\}}$ is the indicator of event occurrence.

B. Cox model

The Cox model is widely used in survival analysis [6]. As a semi-parametric approach, it requires no knowledge of underlying survival distributions and assumes that covariates have the exponential influence on the hazard. For the i-th

individual, the corresponding hazard function specified by the Cox model is given by

$$h(t \mid \mathbf{x}_i) = h_0(t) \exp[g(\mathbf{x}_i)], \tag{2}$$

where $g(\mathbf{x}_i) = \boldsymbol{\beta}^T \mathbf{x}_i$ is the linear relative risk score function, and $\boldsymbol{\beta}$ is the coefficient vector. The baseline hazard function h_0 can be an arbitrary non-negative function of time. In this model, it is infeasible to estimate $\boldsymbol{\beta}$ and the hazard function by using the full likelihood as h_0 is not specified. To deal with this estimation problem, the partial likelihood that does not involve the baseline hazard function is proposed to fit the model [6]. Suppose there are N distinct time points at which the event of interest occurs, denoted by $T_1 < T_2 < \cdots < T_N$, and the covariate vector $\mathbf{x}$ with the same subscript is the corresponding individual. Let R_i be the set of all individuals at risk at time T_i (corresponding to $\min\{T^\star, C\} \geq T_i$). The Cox partial likelihood is given by

$$L_{\text{cox}} = \prod_i \left(\frac{\exp[g(\mathbf{x}_i)]}{\sum_{j \in R_i} \exp[g(\mathbf{x}_j)]} \right)^{D_i}. \tag{3}$$

In practice, the negative partial log-likelihood is used to fit the model:

$$L_{\text{loss}} = -\sum_i D_i \left\{ g(\mathbf{x}_i) - \log[\sum_{j \in R_i} \exp g(\mathbf{x}_j)] \right\}. \tag{4}$$

The cumulative hazard function $H(\cdot)$ can be estimated by the Breslow estimator using $\hat{\boldsymbol{\beta}}$ that minimises L_{loss}.

The classical Cox model is unable to handle high-dimensional features due to the overfitting problem, which encourages the regularised Cox models that take advantage of the norm regularisation to shrink coefficients. Two representative methods are Lasso-Cox with the ℓ_1 norm regularisation and Ridge-Cox with the ℓ_2 norm regularisation [7], [8].

C. Graph convolutional networks

GCNs generalise the convolution operator to the graph domain and achieve promising performance in many areas. To investigate their potential in survival analysis, we adapt the commonly used GCNs proposed in [20] for analysing high-dimensional time-to-event data.

A graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set, and $\mathcal{E}$ is the edge set. In this work, we consider unweighted graphs, and each node in the graph represents a sample with high-dimensional features. Suppose there are n samples and each sample is characterised by a p-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^p$. Let $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ be the adjacency matrix of $\mathcal{G}$, and $D = \text{diag}(d_1, d_2, \ldots, d_n)$ is the diagonal degree matrix where $d_i = \sum_j a_{ij}$ is the degree of node i. The convolution matrix $\hat{A}$ in GCNs is given by $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I_n$, and $\tilde{D} = \text{diag}(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n)$ is the degree matrix calculated from $\tilde{A}$ with $\tilde{d}_i = \sum_j \tilde{a}_{ij}$. We adapt the two-layer GCN proposed in [20] by replacing the last layer with a full-connected layer that outputs the risk scores of samples; that is, the relative risk function $g(\cdot)$ in the Cox model is now parameterised by the GCN instead of a linear function $\boldsymbol{\beta}^T \mathbf{x}_i$. Given the data matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times p}$, and the corresponding $\hat{A}$, the GCN uses the following propagation rule to output the relative risk scores:

$$\mathbf{z} = \text{ReLU}(\hat{A} X W_1)\, W_2, \tag{5}$$

where $W_1 \in \mathbb{R}^{p \times l_h}$ is the weight matrix for the graph convolutional layer, $W_2 \in \mathbb{R}^{l_h \times 1}$ is the weight matrix for the output layer, $\text{ReLU}(\cdot) = \max(0, \cdot)$ is the ReLU activation function, and $\mathbf{z} \in \mathbb{R}^n$ contains the relative risk scores of n samples. With the set of observed times $\{T_1, T_2, \ldots, T_n\}$ and the set of event indicators $\{D_1, D_2, \ldots, D_n\}$, we train the weights by substituting the risk scores in (4) with $\mathbf{z}$ from (5) and optimising L_{loss} with gradient descent.

Unlike those in the original GCNs, the input graph in our model only consists of the training samples during the training phase. After training, for an unseen new sample, we first derive a new graph which consists of both the training samples and the new sample, and then plug the corresponding new adjacency and degree matrices into (5) to get the risk score of the unseen sample.

D. Geometric graphs

Inspired by the fact that Continuous k-Nearest neighbour (CkNN) graphs achieve good performance in node classification with GCNs [25], [26], we feed CkNN graphs into GCNs for performing survival analysis in this paper. For a graph with the number of neighbours for each sample fixed, a pair of samples that lie in a poorly sampled area could be connected in the graph even though they are dissimilar and far away from each other. The edge between these two dissimilar samples is detrimental to the performance of GCNs with the graph as the graph convolution homogenises the features of samples. In contrast, CkNN graphs are able to alleviate this issue by adapting edge densities for different samples. The adjacency matrix $A^{\text{CkNN}} = [a_{ij}^{\text{CkNN}}] \in \mathbb{R}^{n \times n}$ of a CkNN graph is defined as

$$a_{ij}^{\text{CkNN}} = \begin{cases} 1 & \text{if } \text{dist}(i,j) < \delta \sqrt{\text{dist}(i,i_k)\ \text{dist}(j,j_k)}, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where δ is a parameter that regulates the graph density, and $\text{dist}(i, i_k)$ is the Euclidean distance from a sample to its k-th nearest neighbour. The establishment of a link between two samples in a CkNN graph depends on the densities of their regions. To investigate how to appropriately apply CkNN graphs to time-to-event data, we fix $\delta = 1$ and vary k for simplicity. CkNN graphs become sparser/denser with smaller/larger k values. Note that when $k = 0$, A^{CkNN} equals a zero matrix and $\tilde{A}^{\text{CkNN}}$ is an identity matrix; that is, there is no edge between any samples.

E. Evaluation metrics

In this work, we adopt two metrics for evaluating survival models: the time-dependent concordance index and the integrated Brier score, which evaluate the discriminative performance and calibration performance of survival models, respectively.

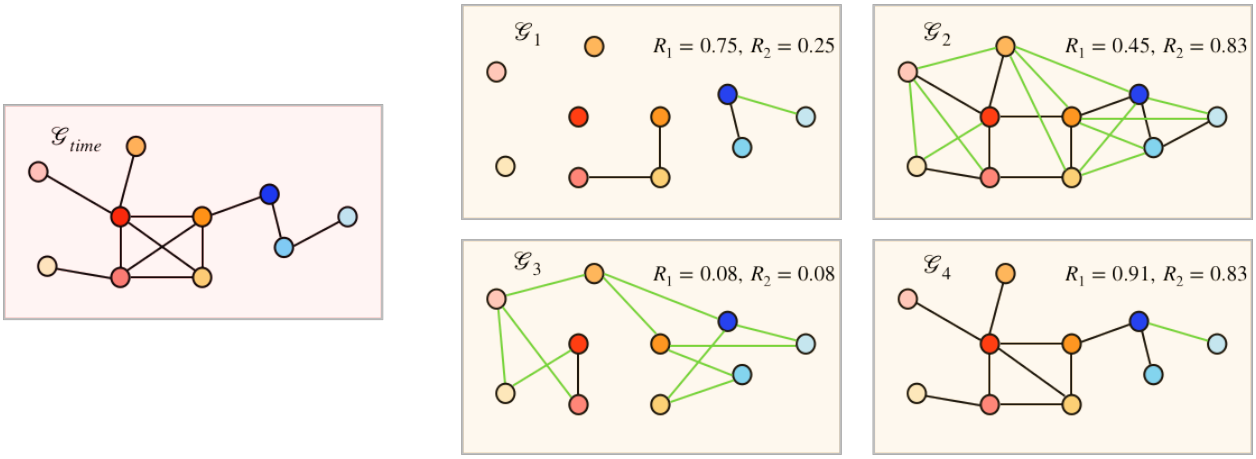


Fig. 1. Illustrative examples of four graphs ($\mathcal{G}_1$-$\mathcal{G}_4$) having different values of R_1 and R_2. The nodes and their spatial locations are the same in different graphs. Edges that are consistent with those in $\mathcal{G}_{\text{time}}$ are shown in black; otherwise, they are shown in green.

1) Time-dependent concordance index: Harrell's concordance index is a widely used metric for evaluating the discrimination ability of survival models [27]. It extends the area under the receiver operating characteristic curve (AUC), which is used for assessing binary classification models [28], to the case of time-to-event prediction. Furthermore, it has been demonstrated that Harrell's concordance index is closely related to the time-specific AUC [29]. Harrell's concordance index of a model that outputs the survival function $\hat{S}(\cdot)$ is defined as

$$\text{CI}_H = P\left(\hat{S}(t \mid \mathbf{x}_i) > \hat{S}(t \mid \mathbf{x}_j)) \mid T_j < T_i, D_j = 1\right), \quad (7)$$

The CI_H evaluates the consistency between the ordering of predicted survival times and that of true survival times. When applied to proportional hazards models, we only need to calculate $g(\cdot)$'s which are independent of t to obtain CI_H since the ranking between the predicted survival times remains the same over time. However, it is inappropriate to apply CI_H to the survival models where the ordering of survival predictions is time-dependent as CI_H does not account for this. Therefore, we adopt the time-dependent concordance index proposed in [30] to account for the ordering of the survival estimates that possibly changes over time. The time-dependent concordance index is given by

$$\text{CI} = P\left(\hat{S}(T_j \mid \mathbf{x}_i) > \hat{S}(T_j \mid \mathbf{x}_j)) \mid T_j < T_i, D_j = 1\right). \quad (8)$$

The CI falls in the range $[0, 1]$, and a CI value closer to 1 is better. Note that CI reduces to CI_H for the proportional hazards models.

2) Integrated Brier score: The Brier score is a measure that assesses the inaccuracy of probabilistic forecast [31]. For the binary classification of n instances with labels $y_i \in \{0, 1\}$, the Brier score of a model that outputs the probability $\hat{P}(y_i = 1 \mid \mathbf{x}_i)$ is formulated as

$$\frac{1}{n}\sum_{i=1}^{n}[\hat{P}(y_i = 1 \mid \mathbf{x}_i) - y_i]^2.$$

The Brier score has been generalised to time-to-event data by calculating the Brier scores for different time points and integrating them [32]. Specifically, for a fixed time t, we get the binary outcomes from time-to-event data in terms of whether the survivals of patients are longer than t or not and measure the calibration at t with predicted survival estimates. The Brier score at t is defined as

$$\text{BS}(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\hat{S}^2(t \mid \mathbf{x}_i)\mathbb{1}(T_i \le t, D_i = 1)}{\hat{G}(T_i)} + \frac{\left(1 - \hat{S}(t \mid \mathbf{x}_i)\right)^2 \mathbb{1}(T_i > t)}{\hat{G}(t)}\right], \quad (9)$$

where $\hat{G}(\cdot)$ is the Kaplan-Meier estimate of the survival function of the censoring distribution $P(C > t)$, and serves as a weighting function for instances. $\text{BS}(t)$ evaluates the calibration ability of a model for a fixed time point. To measure the inaccuracy of survival predictions for a time interval, we consider the integrated BS (IBS):

$$\text{IBS} = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} \text{BS}(s)ds. \quad (10)$$

In practice, we employ numerical integration to approximate IBS. Lower values of IBS suggest better calibration of survival estimates.

III. Methodology

In this section, we first present the motivation of using sparse graphs constructed from high-dimensional features for GCNs. We then propose to use multiple sparse graphs to uncover the local neighbourhoods of samples, which, compared with a single sparse graph, could be more consistent with the survival times of samples. Finally, we introduce a sequential forward floating selection algorithm that yields survival predictions by aggregating information from different graphs with the aid of GCNs.

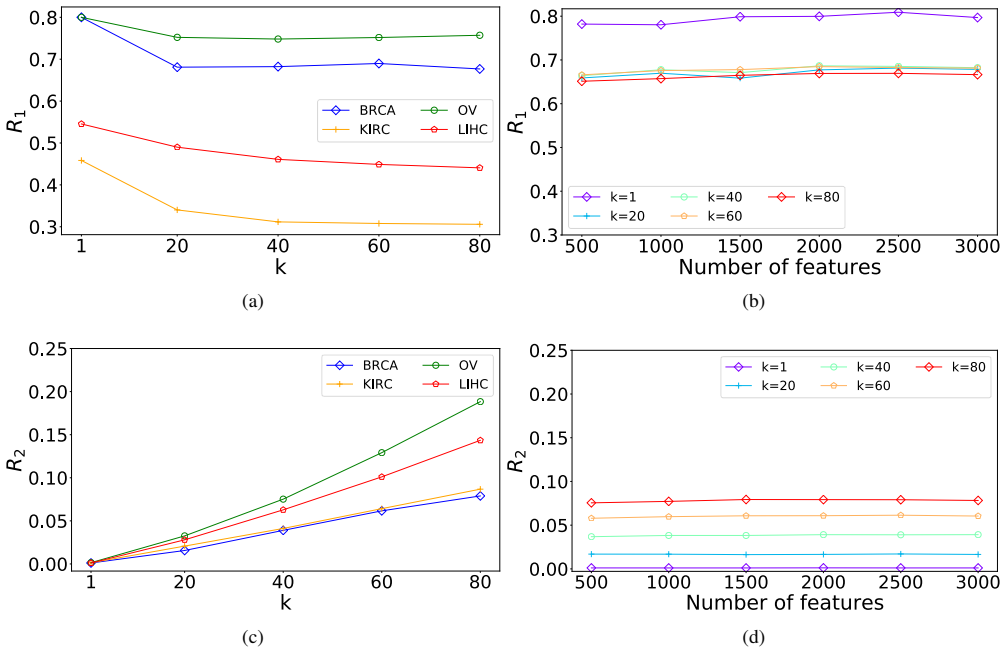


Fig. 2. Plots of R_1 and R_2 calculated from CkNN graphs. R_1 (and (c) R_2) versus k on four high-dimensional survival datasets, whose information is provided in Table I. The corresponding CkNN graphs are built by using all available features and different values of k. (b) R_1 (and (d) R_2) with varying number of sampled features and different k on the BRCA dataset. The corresponding CkNN graphs are constructed from subsets of features. The sampling and graph-construction procedure for a fixed size (number of features) is repeated 10 times and we present their average R_1 (and R_2) values. The threshold c_s is set to the difference between the maximal and the minimal survival times divided by 5 when computing R_1 and R_2.

A. Motivation of using sparse graphs

In this subsection, we investigate how the parameter k of a CkNN graph affects the alignment between the graph and sample survival times, and our finding motivates using sparse CkNN graphs. First, we evaluate the alignment in terms of two criteria that quantify the similarity between the survival times of samples and a CkNN graph. We then find that sparser CkNN graphs with smaller k may be favoured by GCNs for survival analysis.

As discussed before, GCNs with a graph may help survival analysis if the graph structure is consistent with the survival times of samples. That is, two samples connected by an edge in the graph tend to have similar survival times. We therefore evaluate the quality of a geometric graph by comparing the consistency between its edge set and the survival times. For some pairs of samples, we are unable to determine whether there should be edges connecting them based on the difference of their survival times due to censoring. We thus exclude these pairs when comparing the edge set of a graph and the survival times. The set of comparable pairs is defined as

$$\begin{aligned}\mathcal{S}_{com} = \{(i,j) \mid (T_i - T_j) > c_s, D_j = 1 \\ \text{or} \quad |T_i - T_j| \le c_s, D_i = 1, D_j = 1\}\end{aligned} \tag{11}$$

where c_s is a pre-selected threshold for determining whether the survival times of two samples are similar or not. Note that when evaluating the consistency, we only consider the pairs in $\mathcal{S}_{com}$. The set of the pairs of samples with similar survival times $\mathcal{E}_{\text{time}}$ is given by

$$\mathcal{E}_{\text{time}} = \{(i,j) \mid |T_i - T_j| \le c_s, D_i = 1, D_j = 1\}. \tag{12}$$

We denote the edge set of a CkNN graph by $\mathcal{E}_{CkNN}$. The set of edges within the comparable pairs is $\mathcal{E}'_{CkNN} = \mathcal{E}_{CkNN} \cap \mathcal{S}_{com}$. With the principle that the edge set of the graph used in a GCN should be as similar to $\mathcal{E}_{\text{time}}$ as possible, we use the following two criteria to compare $\mathcal{E}'_{CkNN}$ to $\mathcal{E}_{\text{time}}$:

- the ratio of appropriate edges in the CkNN graph:

$$R_1 = \frac{\left|\mathcal{E}_{\text{time}} \cap \mathcal{E}'_{CkNN}\right|}{\left|\mathcal{E}'_{CkNN}\right|};$$

- the fraction of edges in $\mathcal{E}_{\text{time}}$ that have been successfully discovered by the CkNN graph:

$$R_2 = \frac{\left|\mathcal{E}_{\text{time}} \cap \mathcal{E}'_{CkNN}\right|}{\left|\mathcal{E}_{\text{time}}\right|}.$$

Note that R_1 and R_2 are similar to the widely-used precision and recall which are metrics for evaluating classification models [33], respectively, in the sense that the positive instances are now defined as pairs in $\mathcal{E}_{\text{time}}$. The higher R_1 and R_2, the better the CkNN graph aligns with the survival times. Specifically, a high value of R_1 indicates that the corresponding graph tends to connect samples with similar survival times, while higher R_2 means that more pairs of samples with similar survival times are retrieved by the graph. The example graphs in Fig. 1 illustrate the characteristics of graphs with different R_1 and R_2 values. As suggested by the high value of R_1

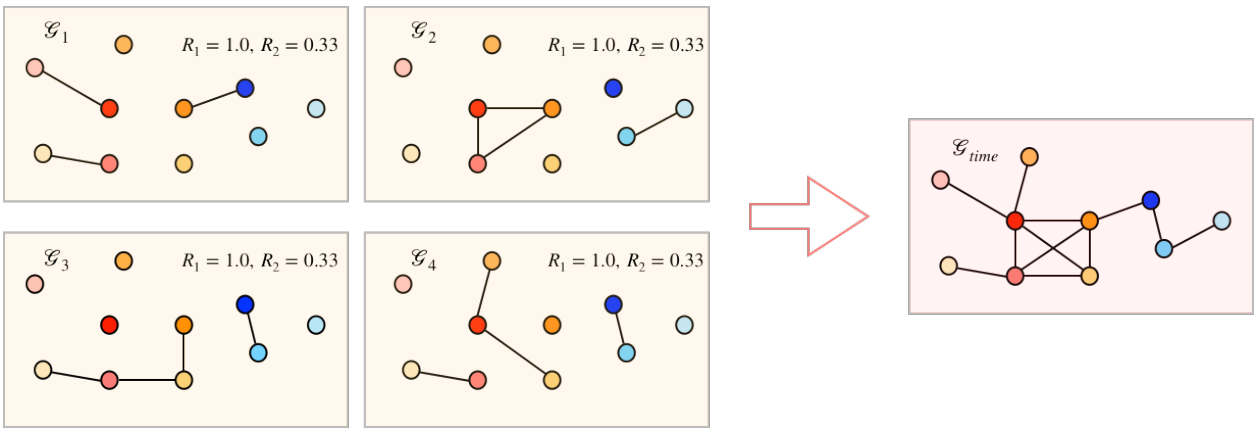


Fig. 3. Illustration of using multiple sparse graphs ($\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$) to uncover a graph ($\mathcal{G}_{\text{time}}$). The nodes and their spatial locations in each graph are the same. The union of the corresponding edge sets with the same subscripts $\bigcup_{i=1}^{4} \mathcal{E}_i = \mathcal{E}_{\text{time}}$.

of $\mathcal{G}_1$, most edges identified by $\mathcal{G}_1$ link samples with similar survival times. But a relatively lower value of R_2 indicates that a number of of edges present in $\mathcal{G}_{\text{time}}$ are missing in $\mathcal{G}_1$, which means that $\mathcal{G}_1$ uncovers only a subset of the edges in $\mathcal{G}_{\text{time}}$. Although $\mathcal{G}_2$ obtains a higher value of R_2 and discovers most edges in $\mathcal{G}_{\text{time}}$, nearly half of the edges detected by the graph do not exist in $\mathcal{G}_{\text{time}}$, which is indicated by a mediocre R_1. Regarding $\mathcal{G}_3$, it is clearly observed that the graph is at odds with $\mathcal{G}_{\text{time}}$, shown by the lowest values of both R_1 and R_2, compared with the other three graphs. Among these four graphs, $\mathcal{G}_4$ fits best with GCNs: the graph structure is highly consistent with $\mathcal{G}_{\text{time}}$ and it achieves high values for both R_1 and R_2.

Fig. 2(a) and Fig. 2(c) present the plots of R_1 and R_2 against varying k on different high-dimensional survival datasets. It is observed that there exists a negative correlation between R_1 and k, while the opposite trend is observed for R_2. The patterns presented in these two figures indicate that sparser graphs are able to achieve higher R_1 while denser graphs can obtain higher R_2. It seems difficult for a single graph to achieve both high R_1 and high R_2. Sparse and dense graphs could be exemplified by $\mathcal{G}_1$ and $\mathcal{G}_2$ in Fig. 1, respectively.

Although dense graphs can uncover more neighbours for a node than sparse graphs, a dense graph with a big k may assign many inappropriate neighbours to nodes. As illustrated by $\mathcal{G}_2$ in Fig. 1, for most nodes, a large fraction of their neighbours have dissimilar survival times. Thus, performing the graph convolution on most nodes could exacerbate their survival predictions. Further, take a limiting case for instance, a GCN with the complete graph ($A = \mathbf{1}\mathbf{1}^T - I$) makes the feature of one sample replaced with the average of all the available samples. It is expected that a survival model based on the GCN with such a graph performs no better than a random predictor since the features of all samples are the same after smoothing. Compared with dense graphs, a GCN with the graph obtained by small k is able to improve survival predictions of most nodes with neighbours. Take the sparse graph ($\mathcal{G}_1$) presented in Fig. 1 for an example, among the nodes having neighbours, four out of five are connected to nodes with similar survival times. Therefore, a GCN with $\mathcal{G}_1$ possibly improves the predictions of these nodes. Based on the above comparison, we reason that sparse graphs are favoured in practice when used in GCNs for survival analysis. Note that the limit of sparse graphs is the empty graph and the corresponding GCN is equivalent to the MLP.

B. Multiple sparse graphs

As discussed before, graphs with both high R_1 and high R_2 fit with GCNs well. Although sparse geometric graphs are able to achieve high R_1, their R_2 values are often small, since a single sparse graph is unlikely to uncover the complete local neighbourhoods of all nodes. We therefore propose to unveil more neighbours for each node and improve R_2 by combining multiple sparse graphs within GCNs. Inspired by the fact that the finite edge set $\mathcal{E}_{\text{time}}$ can be decomposed into subsets, we propose to first construct a set of multiple sparse graphs $S_{\mathcal{G}} = \{\mathcal{G}_i, \ldots, \mathcal{G}_{n_G}\}$, where $\mathcal{G}_i = (\mathcal{V}, \mathcal{E}_i)$ for $i = 1, 2, \ldots, n_G$. A subset of graphs $S_{\text{select}} \subset S_{\mathcal{G}}$ are then chosen such that the aggregation of the corresponding edge sets $\bigcup_{\mathcal{G}_i \in S_{\text{select}}} \mathcal{E}_i$ is more consistent with $\mathcal{E}_{\text{time}}$. Fig. 3 illustrates an ideal case where multiple sparse graphs can be combined to discover all the edges in $\mathcal{E}_{\text{time}}$. All sparse graphs ($\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$) in Fig. 3 have perfect R_1 values but low R_2 values. These graphs are diverse in the sense that their edge sets do not overlap too much. Although none of them fully uncovers $\mathcal{E}_{time}$, their combined graph is the same as $\mathcal{G}_{time}$ and achieves $R_1 = R_2 = 1$. This illustrative example indicates that combining diverse sparse graphs may increase R_2 over original sparse graphs without dramatically decreasing R_1.

The first problem of adopting the proposed approach is how to construct multiple sparse graphs from available features. As two samples could be regarded as being similar in one graph and being dissimilar in another graph when the graphs are constructed using different subsets of features, we propose to use random subsets of features to construct diverse sparse

Algorithm 1 Survival analysis using GCNs and CkNN graphs

Input: the data matrix $X \in \mathbb{R}^{n\times p}$, the set of observed time $\{T_1, T_2, \ldots, T_n\}$, the set of event indicators $\{D_1, D_2, \ldots, D_n\}$

Output: the set of selected graphs S_k, the aggregation model Ridge-Cox(S_k, λ) with the estimated coefficient vector $\mathbf{w}_k$.

1: **Initialisation:** sample subsets of features from X and construct multiple CkNN graphs $S_\mathcal{G} = \{\mathcal{G}_i\}$, where $i = 1, \ldots, n_\mathcal{G}$.
2: **Step 1:** input each subset of features and the corresponding $\mathcal{G}_i$ into a GCN and obtain the trained model M_i, where $i = 1, \ldots, n_\mathcal{G}$.
3: **Step 2:** calculate the risk scores $\{\hat{g}_{ij}\}_{\text{val}}$ of the validation data and $\{\hat{g}_{is}\}_{\text{train}}$ of the training data, where $i = 1, \ldots, n_\mathcal{G}$, $j = 1, \ldots, n_{\text{val}}$, and $s = 1, \ldots, n_{\text{train}}$.
4: **Step 3:** feed $\{\hat{g}_{ij}\}_{\text{val}}$, $\{\hat{g}_{is}\}_{\text{train}}$, and $S_\mathcal{G}$ into Algorithm 2 to obtain the output.

graphs. Fig. 2(b) and Fig. 2(d) show that the values of R_1 and R_2 of the graphs obtained with different random subsets of features do not change much from the graph built with all features. Thus, it is possible to combine these sparse graphs to improve R_2 without decreasing R_1, leading to a more suitable graph for survival analysis.

The second question naturally arising from the above analysis is how to select graphs and combine them for survival analysis. To address this problem, we propose an implicit approach for the selection of graphs with the aid of survival information. Our intuition is that, if the combination of the graphs in S_{select} aligns well with $\mathcal{E}_{\text{time}}$, the model that aggregates the outputs from the GCNs with the graphs in S_{select} should perform well in terms of an evaluation metric for the time-to-event prediction. From this intuition, we select the graphs in such a way that aggregating the corresponding outputs results in a good performance, after training GCNs with different graphs and obtaining their risk scores (Algorithm 2).

C. Algorithms

In the following section, we present our model which **AG**gregates the outputs from **G**CNs for **Surv**ival analysis (AGGSurv) and describe the detailed algorithm of graph selection and aggregation.

1) AGGSurv: We first use random subsets of features to build different CkNN graphs. Inspired by the fact that better alignment between the features and the graph in a GCN benefits the classification performance [23], we then use the subset of features from which the CkNN graph is constructed as the input to the GCN model rather than all the features. After training, for each constructed CkNN graph $\mathcal{G}_i$ with its edge set $\mathcal{E}_i$, we obtain its corresponding survival model M_i. Finally, we select a subset of the constructed graphs and learn the aggregation model simultaneously with the training data and the validation data. The whole algorithm is summarised in Algorithm 1.

Algorithm 2 SFFS algorithm for selecting graphs and learning the aggregation model

Input: the set of CkNN graphs $S_\mathcal{G} = \{\mathcal{G}_i\}$, the corresponding output risk scores $\{\hat{g}_{ij}\}_{\text{val}}$ and $\{\hat{g}_{is}\}_{\text{train}}$, where $i = 1, \ldots, n_\mathcal{G}$, $j = 1, \ldots, n_{\text{val}}$, and $s = 1, \ldots, n_{\text{train}}$.

Output: the set of selected graphs S_k, and the aggregation model Ridge-Cox(S_k, λ) with the estimated coefficient vector $\mathbf{w}_k$.

1: **Initialisation:** $S_0 = \emptyset$, $k = 0$, $\mathbf{w}^+ = [1]$, and $c_{\text{best}} = 0$. We first select the graph $\mathcal{G}^+$ that obtains the highest CI on the validation data and set c to the corresponding CI.
2: **while** $c > c_{\text{best}}$ and $k \leq n_\mathcal{G}$ **do**
3: $S_{k+1} = S_k \bigcup \{\mathcal{G}^+\}$
4: $c_{\text{best}} = c$
5: $k = k + 1$
6: $\mathbf{w}_k = \mathbf{w}^+$
7: #**conditional exclusion step**:
8: **if** $|S_k| > 2$ **then**
9: $\mathcal{G}^- = \underset{\mathcal{G}_j \in S_k}{\arg\max} \left\{ \max_\lambda \text{CI}\left[\text{Ridge-Cox}(S_k \backslash \{\mathcal{G}_j\}, \lambda)\right] \right\}$
10: $\mathbf{w}^-$ = the estimated coefficient vector $\hat{\boldsymbol{\beta}}$ of Ridge-Cox$(S_k \backslash \{\mathcal{G}^-\}, \lambda)$.
11: c_e = CI obtained by Ridge-Cox$(S_k \backslash \{\mathcal{G}^-\}, \lambda)$
12: **if** $c_e > c$ **then**
13: $S_{k-1} = S_k \backslash \{\mathcal{G}^-\}$
14: $c_{\text{best}} = c_e$
15: $\mathbf{w}_{k-1} = \mathbf{w}^-$
16: $k = k - 1$
17: **end if**
18: **end if**
19: $\mathcal{G}^+ = \underset{\mathcal{G}_i \in S_\mathcal{G} \backslash S_k}{\arg\max} \left\{ \max_\lambda \text{CI}\left[\text{Ridge-Cox}(S_k \bigcup \{\mathcal{G}_i\}, \lambda)\right] \right\}$
20: $\mathbf{w}^+$ = the estimated coefficient vector $\hat{\boldsymbol{\beta}}$ of Ridge-Cox$(S_k \bigcup \{\mathcal{G}^+\}, \lambda)$.
21: c = CI obtained by Ridge-Cox$(S_k \bigcup \{\mathcal{G}^+\}, \lambda)$
22: **end while**

2) SFFS: After training the multiple GCNs with different CkNN graphs, we propose a sequential forward floating selection (SFFS) algorithm to select a subset of the constructed graphs and employ a survival model that learns how to best aggregate the predictions from the GCNs trained with the selected graphs (Algorithm 2). For the aggregation model, we adopt the Ridge-Cox model because the sizes of the datasets are small and using other survival models which require large training data may exacerbate the overfitting problem. Furthermore, the coefficients of the Ridge-Cox model can reflect the impact of each selected graph on the final predictions. Note that the input features to the Ridge-Cox model are the output predicted risk scores from the trained GCNs.

Let S_k be a set of k selected graphs ($|S_k| = k$). Steps 3-6 and 19-21 in Algorithm 2 show the process of including a new graph in S_k, and steps 9-16 in Algorithm 2 present the process of excluding a graph from S_k. The SFFS algorithm includes a graph $\mathcal{G}_i$ in S_k or excludes a graph $\mathcal{G}_j$ from S_k in terms of the concordance index

(CI) obtained by the model Ridge-Cox($S_k \bigcup \{\mathcal{G}_i\}, \lambda$)) or Ridge-Cox($S_k \setminus \{\mathcal{G}_j\}, \lambda$)). Specifically, we first train a Ridge-Cox model to learn how to best aggregate the predictions from the GCNs with the graphs in $S_k \bigcup \{\mathcal{G}_i\}$ or $S_k \setminus \{\mathcal{G}_j\}$ on the training data. The regularisation parameter λ of the Ridge-Cox model is selected such that the Ridge-Cox model achieves the best discriminative performance (CI) on the validation data, which is $\max_{\lambda}$ CI [Ridge-Cox($S_k \setminus \{\mathcal{G}_j\}, \lambda$)] for the exclusion step and $\max_{\lambda}$ CI [Ridge-Cox($S_k \bigcup \{\mathcal{G}_i\}, \lambda$)] for the inclusion step. This process is iterated for all the graphs that can be possibly added to or removed from the set S_k, and the one with the highest CI on the validation data is added or removed. The algorithm stops when the CI cannot be increased anymore. Note that the output selected graphs in S_k can be combined into a weighted graph, where the edge set is the union of those of selected graphs and the weight for each edge can be obtained by the weighted sum of the different graphs with the final weight vector $\mathbf{w}_k$ of the Ridge-Cox model.

IV. Experiments

In this section, the proposed AGGSurv model is extensively evaluated with comparisons to several other methods. Nine high dimensional datasets are used in our experiments. The results show that AGGSurv outperforms the other approaches.

A. Datasets

The comparison of survival analysis performance is conducted on a variety of high-dimensional survival datasets, including eight datasets from The Cancer Genome Atlas (TCGA) (https://www.cancer.gov/tcga) and one dataset from the TADPOLE challenge (https://tadpole.grand-challenge.org). The datasets from TCGA are downloaded by using the R/Bioconductor package: RTCGAToolbox [34]. The TCGA datasets include the following eight cancer types: breast invasive carcinoma (BRCA), kidney renal clear cell carcinoma (KIRC), lung adenocarcinoma (LUAD), urothelial bladder carcinoma (BLCA), head and neck squamous cell carcinoma (HNSC), brain lower grade glioma (LGG), liver hepatocellular carcinoma (LIHC), and ovarian serous cystadenocarcinoma (OV). The event of interest is defined as death for cancer patients in these datasets. The high-dimensional features that we extract from TCGA for performing time-to-event prediction are the normalised RNA sequencing data. The dataset from the TADPOLE challenge is obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [35] and it contains patients with mild cognitive impairment (MCI) at baseline having a substantially increased risk of developing Alzheimer's disease (AD). The event of interest is defined as MCI to AD conversion for patients with MCI. We preprocess the dataset from TADPOLE by following the procedure presented in [36]. The features used for predicting the conversion include the MRI features, Apolipoprotein E4, and gender.

The characteristics of these datasets are summarised in Table I. In the pre-processing of the TCGA datasets, we remove the features with 0 variance and add a pseudocount 1 to all features, followed by a log transformation. We then standardise the log-transformed data to make them centered around 0 with a unit variance for each feature.

TABLE I
SUMMARY OF THE DATASETS USED IN THE EXPERIMENTS.

Dataset	# patients	# features	Prop. Censored
BRCA	1079	20224	0.860
KIRC	531	20221	0.670
LUAD	503	20172	0.638
BLCA	405	20215	0.560
HNSC	519	20234	0.576
LGG	511	20199	0.755
LIHC	365	20140	0.644
OV	303	20161	0.393
AD	439	252	0.426

TABLE II
HYPERPARAMETERS SEARCH SPACES.

Hyperparameter	Values
# Nodes in the hidden layer	$16, 32, 64$
Batch size	$128, 256$
k (GSurv)	$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$
# durations (DeepHit)	$10, 20$
α (DeepHit)	$0, 1$
σ (DeepHit)	$0.1, 0.5, 1, 2.5, 5, 10, 100$
Split rule (RSF)	log rank, log rank score
Node size (RSF)	$5, 10, 15, 20$

B. Methods

We compare AGGSurv to the classical Lasso-Cox [7], Ridge-Cox [7], [8], and RSF [9], as well as the methods based on neural networks: DeepSurv [2], DeepHit [17], and CoxTime [16]. To investigate whether the combination of multiple sparse graphs is superior to using a single graph, we also take into account the survival model based on the GCN with a single graph (GSurv) where k in the CkNN graph is determined by a grid search over the validation dataset.

C. Implementation details

For evaluation, we apply 10-fold cross validation for each dataset: we randomly separate the data into a training set (90%) and a test set (10%), and 10% of the training set is used as the validation set. For all sets, we keep the same ratio of censoring as in the original datasets.

All (graph) neural networks used in DeepSurv, CoxTime, DeepHit, GSurv, and AGGSurv consist of an input layer, a hidden layer and an output layer. The networks are trained by back-propagation with the Adam optimizer of a learning rate of 5e-4. The dropout probability of 0.1 and the weight decay of 1 are applied. Early stopping is performed based on the validation loss to avoid overfitting.

The optimisation of the other hyperparameters is performed individually for each fold by a grid search, and the configuration is selected such that the corresponding model achieves the best discriminative performance (CI) on the validation set. The search spaces for the hyperparameters are provided in Table II.

TABLE III
COMPARISON OF DIFFERENT APPROACHES IN TERMS OF CI (AVERAGED OVER THE 10 FOLDS). THE RIGHTMOST COLUMN SHOWS THE AVERAGE RANKING IN TERMS OF CI FOR EACH METHOD OVER THE EIGHT DATASETS. NOTE THAT THE LOWER THE RANKING, THE BETTER THE PERFORMANCE. TOP THREE APPROACHES FOR EACH DATASET ARE IN BOLD.

Methods	BRCA	KIRC	LUAD	BLCA	HNSC	LGG	LIHC	OV	AD	Avg. ranking
CoxTime	0.637	0.711	**0.592**	0.636	0.598	0.781	0.652	0.570	0.618	6.000
DeepHit	0.643	0.664	0.581	0.633	0.562	0.759	0.667	**0.582**	0.625	6.444
DeepSurv	**0.715**	**0.723**	**0.599**	**0.644**	0.610	**0.852**	**0.674**	**0.594**	0.687	**2.889**
GSurv	0.696	**0.723**	**0.600**	**0.643**	**0.614**	**0.847**	0.671	0.577	0.696	**3.111**
AGGSurv	**0.719**	**0.721**	**0.592**	**0.664**	**0.621**	**0.852**	**0.693**	**0.605**	**0.697**	**1.778**
RSF	0.581	0.681	0.585	0.622	0.610	0.828	0.656	0.566	**0.702**	5.778
Ridge-Cox	**0.700**	0.709	0.585	0.612	**0.632**	0.843	**0.691**	**0.582**	0.690	4.111
Lasso-Cox	0.660	0.691	0.573	0.619	0.600	0.841	0.596	0.569	**0.722**	5.889

TABLE IV
COMPARISON OF DIFFERENT APPROACHES IN TERMS OF IBS (MEAN VALUE OVER THE 10 FOLDS). THE TOP THREE OF EACH DATASET ARE IN BOLD.

Methods	BRCA	KIRC	LUAD	BLCA	HNSC	LGG	LIHC	OV	AD	Avg. ranking
CoxTime	**0.152**	**0.168**	0.243	0.220	0.246	**0.157**	0.208	**0.186**	0.205	**3.667**
DeepHit	0.168	0.198	**0.237**	0.238	0.239	0.177	**0.201**	0.220	0.208	5.222
DeepSurv	**0.154**	**0.169**	0.246	0.239	0.241	0.169	0.232	0.189	**0.184**	4.556
GSurv	0.177	0.169	0.263	0.240	0.243	0.186	0.235	0.207	0.185	6.333
AGGSurv	**0.161**	**0.163**	**0.228**	**0.214**	**0.222**	**0.163**	**0.205**	**0.176**	0.196	**2.222**
RSF	0.167	0.178	**0.215**	**0.218**	**0.223**	0.169	**0.207**	**0.173**	**0.178**	**2.778**
Ridge-Cox	0.168	0.176	0.248	**0.219**	**0.238**	**0.163**	0.213	0.187	**0.184**	4.111
Lasso-Cox	0.205	0.200	0.306	0.239	0.247	0.229	0.249	0.223	**0.175**	7.111

The search spaces listed in the top two rows in Table II are applied to all methods based on the (graph) neural networks. Lasso-Cox and Ridge-Cox are implemented with the R package glmnet [8] and the regularisation parameters are selected from the sequence provided by the package. The Ridge-Cox model used in the aggregation process is implemented with the python scikit-survival library [37] and the search space for the regularisation parameter λ is $[1e+3, 1e+4, 1e+5, 1e+6]$. The number of trees in RSF is 500, which is found to be empirically sufficient.

It is possible that samples (nodes) are too far away from each other, in which case considering the relation between nodes using GCNs is unnecessary, particularly for datasets of a small size. We thus take into account the case of an empty graph when implementing GSurv and AGGSurv, which is equivalent to DeepSurv. Note that the inclusion of an empty graph would not change the graph structure obtained by combining the selected graphs in the sense that the edge set of empty graph ($\tilde{A} = I$) is an empty set and $\bigcup_i \mathcal{E}_i \bigcup \varnothing = \bigcup_i \mathcal{E}_i$. For each configuration of the network architecture of AGGSurv, we first construct multiple graphs from random subsets of features and train the GCNs to get the corresponding results. Then, we input these results into Algorithm 2 to get the aggregated predictions. Two different sizes of the subset of features for constructing multiple CkNN graphs are used, which are $\{1500, 3000\}$ for the TCGA datasets and $\{100, 200\}$ for the AD dataset. The sampling for each size is repeated 4 times for each configuration; that is, 8 graphs is constructed for each configuration. The parameter k for CkNN is set to 1 so as to build sparse CkNN graphs for AGGSurv.

D. Results and analysis

In the following, we evaluate the discriminative performances of the survival models in terms of CI and the calibration performance according to IBS.

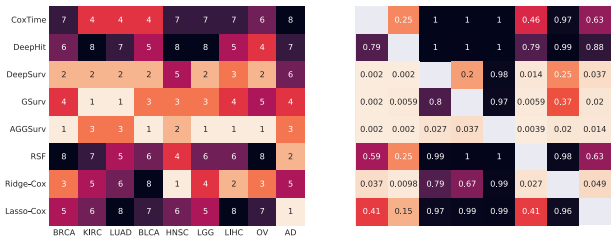


Fig. 4. Visualised comparison of different approaches in terms of CI. Left: visualised rankings of each method on different datasets in terms of CI. The value at the (i, j) position is the ranking of the method in the i-th row on the dataset in the j-th column. Right: visualised p-values of the one-sided Wilcoxon signed-rank test between pairs of approaches obtained by comparing their CI. The alternative hypothesis is that the performance of the method in the i-th row is better than that in the j-th row according to CI or IBS over all the datasets used. Here we set the significance level to 0.05, i.e., one approach is significantly better than the other if the corresponding p-value is smaller than the significance level. Note that brighter colour and lower ranking value indicate better performance.

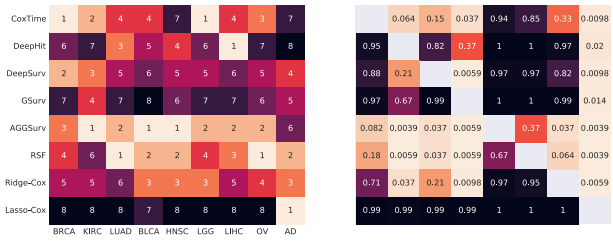


Fig. 5. Visualised comparison of different approaches in terms of IBS. Left: visualised rankings of each method on different datasets in terms of IBS. Right: visualised p-values of the one-sided Wilcoxon signed-rank test between pairs of approaches obtained by comparing their IBS.

Table III shows the mean values of CI obtained by different methods over 10-folds on each dataset and Table IV presents the average IBS. To facilitate the comparison of different approaches, we provide the visualisations of both model rankings and p-values of the one-sided Wilcoxon signed-rank test that tests whether one approach is significantly better than another one in Fig 4 and Fig 5.

We first compare different approaches from the perspective of the discrimination (CI). Ridge-Cox is significantly better than CoxTime, DeepHit, RSF, and Lasso-Cox. On the contrary, DeepSurv and GSurv, which are based on the proportional hazards assumption, perform equally well as Ridge-Cox according to the corresponding statistical testing. It is clear to see that AGGSurv presents the lowest average ranking and its discriminative performance is significantly superior to the other methods.

Second, we compare the methods based on their calibration performances (IBS). AGGSurv performs significantly better than most approaches except for CoxTime and RSF. Furthermore, AGGSurv is superior to CoxTime and RSF according to their average rankings. Although most approaches based on the neural networks and the Cox partial likelihood do rather poorly, AGGSurv generally performs well over the datasets.

Third, by comparing AGGSurv and GSurv, we note that combining multiple sparse graphs does provide an advantage

over a single graph. Furthermore, AGGSurv is shown able to remedy the overfitting problem of high-dimensional datasets.

To sum up, AGGSurv achieves the best discriminative performance and performs well according to the calibration of its survival predictions. Hence, we can conclude that AGGSurv provides a robust approach to survival analysis of high-dimensional datasets by aggregating multiple sparse graphs.

E. Computational complexity

AGGSurv involves two steps: training $n_{\mathcal{G}}$ GCNs with different graphs, and selecting and combining the GCNs with the SFFS algorithm. During the first step, the computational complexity for computing the risk scores with GCNs in (5) is $\mathcal{O}\left(|\mathcal{E}|\, p + n(p+1)l_h\right)$, as the graphs used in AGGSurv are sparse and $\hat{A}X$ can be implemented by sparse-dense matrix multiplications. The computational complexity for evaluating the negative partial log-likelihood function in (4) is $\mathcal{O}(n^2)$. Linear complexity can be achieved by approximating the full risk sets with the sampled risk sets of a fixed size, which is proposed in [16]. During the second step, the SFFS algorithm takes $\mathcal{O}(n_{\mathcal{G}}^2)$ calls of implementing Ridge-Cox to find the subset of the constructed graphs (GCNs). Note that the number of graphs is small, and running the second step is much faster than training the GCNs in practice.

V. Conclusion and Future Work

In this work, we first clarify that the prerequisite for a GCN model to improve survival analysis is to input a graph that aligns well with the sample survival times. With this insight, we propose to combine multiple sparse graphs to uncover a graph where the edges connect samples with similar survival times. We then propose a survival model that not only outputs the survival predictions but also captures the local neighbourhoods, by using multiple sparse graphs for GCNs. The key idea of the proposed approach is to aggregate the information of local neighbourhoods from different sparse graphs and assess the aggregated predictions by the survival information. The experimental results show that the proposed model achieves the best concordance and performs well in terms of its calibration performance.

Two criteria R_1 and R_2 have been used to quantify the alignment between a graph and the survival times of samples. As future works, it would be interesting to investigate how to derive a single criterion, through properly combining R_1 and R_2, to directly find an optimal graph. In the proposed model, we have used the GCN to process the data on the graph. Recently, many new graph neural networks have been proposed, such as Graph Isomorphism Network and Graph Attention Network [21], [38]. In the future, these new architectures can be explored to enhance the calibration performance of the proposed model.

References

[1] R. Li, J. Yao, X. Zhu, Y. Li, and J. Huang, "Graph CNN for survival analysis on whole slide pathological images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 174–182.

[2] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, no. 1, p. 24, Feb 2018.

[3] L. Dirick, G. Claeskens, and B. Baesens, "Time to default in credit scoring using survival analysis: a benchmark study," *Journal of the Operational Research Society*, vol. 68, no. 6, pp. 652–665, Jun 2017.

[4] A. L. Spivak and K. R. Damphousse, "Who returns to prison? A survival analysis of recidivism among adult offenders released in oklahoma, 1985–2004," *Justice Research and Policy*, vol. 8, no. 2, pp. 57–88, 2006.

[5] P. Wang, Y. Li, and C. K. Reddy, "Machine learning for survival analysis: A survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–36, Feb 2019.

[6] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972.

[7] R. Tibshirani, "The lasso method for variable selection in the cox model," *Statistics in Medicine*, vol. 16, no. 4, pp. 385–395, 1997.

[8] N. Simon, J. H. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for Cox's proportional hazards model via coordinate descent," *Journal of Statistical Software, Articles*, vol. 39, no. 5, pp. 1–13, 2011.

[9] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer *et al.*, "Random survival forests," *Annals of Applied Statistics*, vol. 2, no. 3, pp. 841–860, 2008.

[10] F. M. Khan and V. B. Zubek, "Support vector regression for censored data (SVRc): A novel tool for survival analysis," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 863–868.

[11] V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. Suykens, "Support vector methods for survival analysis: a comparison between ranking and regression approaches," *Artificial Intelligence in Medicine*, vol. 53, no. 2, pp. 107–118, 2011.

[12] P. K. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007, pp. 655–660.

[13] C. Liu, W. Cao, S. Wu, W. Shen, D. Jiang, Z. Yu, and H.-S. Wong, "Asymmetric graph-guided multitask survival analysis with self-paced learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[14] W. Gu, Z. Zhang, X. Xie, and Y. He, "An improved muti-task learning algorithm for analyzing cancer survival data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 2, pp. 500–511, 2021.

[15] T. Ching, X. Zhu, and L. X. Garmire, "Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data," *PLoS Computational Biology*, vol. 14, no. 4, p. e1006076, 2018.

[16] H. Kvamme, Ørnulf Borgan, and I. Scheel, "Time-to-event prediction with neural networks and Cox regression," *Journal of Machine Learning Research*, vol. 20, no. 129, pp. 1–30, 2019.

[17] C. Lee, W. Zame, J. Yoon, and M. van der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[18] C. Lee, J. Yoon, and M. Van Der Schaar, "Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 1, pp. 122–133, 2019.

[19] K. Ren, J. Qin, L. Zheng, Z. Yang, W. Zhang, L. Qiu, and Y. Yu, "Deep recurrent survival analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4798–4805.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[22] H. Gao and S. Ji, "Graph U-nets," in *International Conference on Machine Learning*, 2019, pp. 2083–2092.

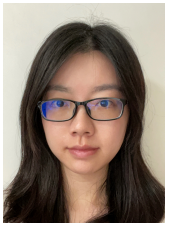
[23] Y. Qian, P. Expert, T. Rieu, P. Panzarasa, and M. Barahona, "Quantifying the alignment of graph and features in deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2021.

[24] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[25] Y. Qian, P. Expert, P. Panzarasa, and M. Barahona, "Geometric graphs from data to aid classification tasks with graph convolutional networks," *Patterns*, vol. 2, no. 4, p. 100237, 2021.

[26] T. Berry and T. Sauer, "Consistent manifold representation for topological data analysis," *Foundations of Data Science*, vol. 1, no. 1, pp. 1–38, 2019.

[27] F. E. Harrell Jr, K. L. Lee, and D. B. Mark, "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in Medicine*, vol. 15, no. 4, pp. 361–387, 1996.
[28] D. Bamber, "The area above the ordinal dominance graph and the area below the receiver operating characteristic graph," *Journal of Mathematical Psychology*, vol. 12, no. 4, pp. 387–415, 1975.
[29] P. J. Heagerty and Y. Zheng, "Survival model predictive accuracy and ROC curves," *Biometrics*, vol. 61, no. 1, pp. 92–105, 2005.
[30] L. Antolini, P. Boracchi, and E. Biganzoli, "A time-dependent discrimination index for survival data," *Statistics in Medicine*, vol. 24, no. 24, pp. 3927–3944, 2005.
[31] G. W. Brier *et al.*, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.
[32] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in Medicine*, vol. 18, no. 17-18, pp. 2529–2545, 1999.
[33] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
[34] M. K. Samur, "RTCGAToolbox: a new tool for exporting TCGA Firehose data," *PloS one*, vol. 9, no. 9, p. e106397, 2014.
[35] R. C. Petersen, P. Aisen, L. A. Beckett, M. Donohue, A. Gamst, D. J. Harvey, C. Jack, W. Jagust, L. Shaw, A. Toga *et al.*, "Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization," *Neurology*, vol. 74, no. 3, pp. 201–209, 2010.
[36] J. Orozco-Sanchez, V. Trevino, E. Martinez-Ledesma, J. Farber, and J. Tamez-Peña, "Exploring survival models associated with MCI to AD conversion: A machine learning approach," *bioRxiv*, p. 836510, 2019.
[37] S. Pölsterl, "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn," *Journal of Machine Learning Research*, vol. 21, no. 212, pp. 1–6, 2020.
[38] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019.



Yurong Ling received the B.S. degree in mathematics from Hunan University in 2014, and the M.Sc. degree in financial mathematics from the London School of Economics and Political Science in 2015. She is currently pursuing the Ph.D. degree in the Department of Statistical Science, University College London. Her research interests include statistical pattern recognition and image processing.

Zijing Liu received his B.Eng. in biomedical engineering from Tsinghua University, M.Sc. in biocomputing from Technical University of Munich and PhD in data science from Imperial College London, where he is now an academic visitor in the Department of Mathematics. He was a research associate in the UK dementia research institute for two years. His research interests include unsupervised machine learning methods, high-dimensional data analysis and their applications in biomedical data.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor of Statistical Pattern Recognition in the Department of Statistical Science at University College London. His research interests include statistical pattern recognition, machine learning, and computer vision. He is an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Cybernetics, and the IEEE Transactions on Neural Networks and Learning Systems.