GraphLSurv: A Scalable Survival Prediction Network with Adaptive and Sparse Structure Learning for Histopathological Whole-Slide Images ⋆ ,⋆⋆

Pei Liu[a], Luping Ji[a], Feng Ye[b], Bo Fu[a,*]

[a]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Xiyuan Ave, Chengdu, 611731, Sichuan, China
[b]Institute of Clinical Pathology, West China Hospital, Sichuan University, Guo Xue Xiang, Chengdu, 610041, Sichuan, China

Abstract

Background and Objective: Predicting patients' survival from gigapixel Whole-Slide Images (WSIs) has always been a challenging task. To learn effective WSI representations for survival prediction, existing deep learning methods have explored utilizing graphs to describe the complex structure inner WSIs, where graph node is respective to WSI patch. However, these graphs are often densely-connected or static, leading to some redundant or missing patch correlations. Moreover, these methods cannot be directly scaled to the very-large WSI with more than 10,000 patches. To address these, this paper proposes a scalable graph convolution network, GraphLSurv, which can efficiently learn adaptive and sparse structures to better characterize WSIs for survival prediction.
Methods: GraphLSurv has three highlights in methodology: (1) it generates adaptive and sparse structures for patches so that latent patch correlations could be captured and adjusted dynamically according to prediction tasks; (2) based on the generated structure and a given graph, GraphLSurv further aggregates local microenvironmental cues into a non-local embedding using the proposed hybrid message passing network; (3) to make this network suitable for very large-scale graphs, it adopts an anchor-based technique to reduce theorical computation complexity.
Results: The experiments on 2,268 WSIs show that GraphLSurv achieves a concordance-index of 0.66132 and 0.68348, with an improvement of 3.79% and 3.41% compared to existing methods, on NLST and TCGA-BRCA, respectively.
Conclusions: GraphLSurv could often perform better than previous methods, which suggests that GraphLSurv could provide an important and effective means for WSI survival prediction. Moreover, this work empirically shows that adaptive and sparse structures could be more suitable than static or dense ones for modeling WSIs.

Keywords: Whole Slide Image, Computational Pathology, Survival Prediction, Graph Convolution Network.

1. Introduction

Histopathological Whole-Slide Image (WSI) is one kind of medical pathology image. It is usually used by pathologists to diagnose several complex tumor diseases. In light of the histopathological features of WSIs, such as tumor invasion, mitoses, anaplasia, and necrosis, clinical doctors can make critical decisions on disease treatment [1, 2]. However, such manual interpretation is often subjective, suffering from large inter- and intra-observer variability. And even patients with the same histopathological features still have distinct survival outcomes due to tumor heterogeneity. Although several classical pathology-omics biomarkers (*e.g.*, TNM stage) have been successfully developed for cancer prognosis, most of them are solely built upon the characteristic of tumorous cells. It is still not enough to completely reveal survival risk as hidden tumor microenvironment also plays an important role in tumor invasion [3, 4, 5].

In recent years, deep learning has achieved considerable success in WSI analysis, such as carcinoma subtype classification [6, 7], tumor stage prediction [8], and prognostic risk estimation [9, 1]. Owing to the strong representation learning ability, deep learning-based networks can automatically extract effective features from WSIs for downstream prediction. Consequently, they greatly alleviate the aforementioned problems posed by subjective human interpretation and inadequate tumor assessment criteria.

Unlike natural images, WSIs usually have typical analysis procedures in deep learning modeling due to their extremely-high resolution (*e.g.*, 150,000 × 150,000 pixels). A common workflow often contains tissue segmentation, WSI patching, patch sampling, patch feature extracting, and WSI-level modeling [10]. This workflow achieves WSI-level prediction by training patch classifiers and then aggregating patch-level responses. However, in clinical practice, obtaining large-scale patch-level annotations are almost impractical for gigapixel WSIs, while WSI-level annotations are much easier to derive from pathology reports. Therefore, only with weak WSI-level labels, how to efficiently learn from unlabeled image patches (may count

⋆This work was supported by Science and Technology Department of Sichuan Province of China under grant 2021YFS0236 and 23QYCX0070, and supported by 1.3.5 project for disciplines of excellence, West China Hospital, Sichuan University under grant ZYJC21035.

⋆⋆Source code is available at https://github.com/liupei101/GraphLSurv for facilitating further research.

*Corresponding author: B. Fu (fubo@uestc.edu.cn).
Email addresses: yuukilp@163.com (Pei Liu), jiluping@uestc.edu.cn (Luping Ji), fengye@scu.edu.cn (Feng Ye), fubo@uestc.edu.cn (Bo Fu)

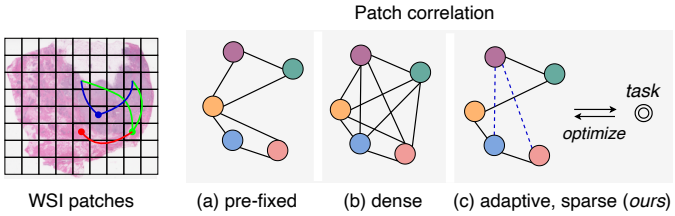

Figure 1: Patch structure is often pre-fixed or dense in existing WSI survival prediction networks. This paper proposes a structure learning method to construct adaptive and sparse patch correlations, which shows to be more suitable for WSI survival prediction.

up to 10,000) and aggregate local information into global representation is a primary problem to tackle in WSI modeling. This problem is often cast as weakly-supervised multiple instance learning (MIL) [11], in which WSI and patches are taken as bag and instances, respectively.

To tackle this problem, many works have proposed different MIL-based solutions for WSI-based survival analysis. These solutions can be roughly classified into two categories. The one is instance-level approach. It trains instance-level models using local samples. And the labels of these local samples are explicitly inherited from WSI-level label [12, 13, 14]. However, these instance-level models often perform not well, because the WSI-level labels tagged on local samples are often inevitably noisy and those predicted local responses could be biased as well [15]. The other one is embedding-level approach, *i.e.*, embedding-based weakly-supervised learning. It trains a global bag-level model to directly predict WSI responses in an end-to-end manner, no longer relying on the noisy label of local samples. This category thus could often achieve better results than the first category [16]. This paper will mainly study on embedding-level solutions.

Embedding-level approaches usually focus on how to construct proper patch correlations to better learn patch embeddings, so as to derive effective representations from WSIs. As representatives, MILSurv [17] and its improved version [18] (DeepAttnMISL) first introduce cluster concepts into patch correlation construction. Different from instance-level approaches, they aggregate cluster-level features into WSI representations. By contrast, DeepGraphSurv [19], RankSurv [20], and PatchGCN [21] employ graph topology to describe patch structure, where a patch embedding is only updated by its connected patches. All of them have achieved promising results, confirming the potential of adopting graphs in learning WSI representations. However, whether patches are organized as clusters or graphs, the patch correlations in most of them are pre-fixed, as illustrated in Figure 1(a), which means the requirement of manual fine-tuning regarding prediction tasks. In addition, for some other methods like [19], patches are often densely-connected, as illustrated in Figure 1(b), which could incur noisy correlations to WSI graphs.

Given the problems posed by pre-fixed or densely-connected patch structures, this paper aims to explore more suitable structures (see Figure 1(c)) to capture the latent patch correlations in WSIs, so as to better characterize gigapixel WSIs for survival prediction. To this end, we propose a novel weakly-supervised embedding-level approach, named GraphLSurv. Specifically, the proposed approach first collects patch-level raw features through WSI preprocessing and then learns how to optimally connect local patches by the proposed structure learning method. Finally, it integrates local microenvironmental cues into WSI-level representation via the proposed GCN-HMP (Graph Convolution Network with Hybrid Message Passing) network.

Our main contributions are summarized as follows:

- A comprehensive framework (GraphLSurv) is developed for WSI survival prediction. It is based on a weakly-supervised learning approach that no longer relies on any local labels and directly makes patient-level predictions.
- We propose a structure learning method to capture potential patch correlations and generate adaptive and sparse structures. It could enable us to optimally select patches for feature aggregation and effectively train GCN without the need to predefine a fixed graph regarding the task.
- A scalable GraphLSurv is further developed for very large-scale WSIs by adopting an anchor-based technique. Algorithm complexity analysis shows that this scalable version can reduce the time and space complexity from $O\left(s^2p\right)$ to $O\left(srp\right)$ and $O\left(s^2\right)$ to $O\left(sr\right)$, respectively, where $r \ll s$.
- The experiments on two publicly-available datasets, NLST and TCGA-BRCA, show that our GraphLSurv could often outperform existing methods. Our ablation study suggests that adaptive and sparse structures could be more suitable to describe the patch correlations in WSIs for survival prediction.

2. Related work

2.1. Survival Analysis

2.1.1. Survival Data

The survival data used for analysis and modeling often includes individual characteristics, follow-up time $t \in \mathbb{R}$, and follow-up status $\sigma \in \{0, 1\}$. In this study, we denote the materials of n patients by a set $\{(x_i, t_i, \sigma_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ is the characteristic of i-th patient; t_i is the observation time of i-th patient; and σ_i is the event status at time t_i. If an event does not occur, then $\sigma = 0$, otherwise $\sigma = 1$. The patient with $\sigma = 0$ is regarded as right-censored and is also included in this study.

2.1.2. Survival Modeling

CoxPH [27], which estimates the hazard of the event under a hypothesis of individual proportional hazard, is the most popular model in survival analysis. The hazard estimation can be expressed by the formulation $\hat{y}_i\left(x_i\right) = f\left(x_i; \theta\right)$, where f denotes a CoxPH model, and θ is model parameter. The estimation error of CoxPH is often minimized by optimizing a partial likelihood

Table 1: Related deep learning methods for WSIs.

Method	Data Structure			Network		Task
	Type	Adaptive	Sparse	Backbone	Scalable	
WSISA (2017) [13]	patch → cluster		✓	CNN		survival
DeepAttnMISL (2020) [18]	patch → cluster		✓	CNN		survival
BDOCOX (2021) [14]	patch → cluster		✓	CNN		survival
DeepGraphSurv (2018) [19]	patch → graph	✓		SpectralGCN		survival
RankSurv (2020) [20]	patch → graph		✓	HyperGCN		survival
PatchGCN (2021) [21]	patch → graph		✓	GCN		survival
DT-MIL (2021) [22]	patch	✓		Self-Attention		classification
DSMIL (2021) [23]	patch	✓		Self-Attention		classification
TransMIL (2021) [14]	patch	✓		Self-Attention		classification
StoHisNet (2022) [24]	patch	✓		Self-Attention		classification
Constrained-MIL (2022) [25]	patch	-	-	Attention MIL	✓	classification
HIPT (2022) [26]	patch hierarchy	✓		Transformer	✓	both
GraphLSurv (ours)	patch → graph	✓	✓	GCN	✓	survival

function. A negative log expression of this likelihood function is written as

$$\ell_{\text{cox}} = \sum_{i\in\{i:\sigma_i=1\}} \left(\hat{y}_i - \log \sum_{j\in\{j:t_j\geq t_i\}} e^{\hat{y}_j} \right). \tag{1}$$

2.1.3. Evaluation Metrics

The evaluation metrics commonly used for survival models is Concordance Index (C-Index) [28]. It mainly measures the ability that survival models could order individual survival risks, which expects a higher risk prediction if the patient dies earlier. We write it by

$$\text{C-Index} = \frac{1}{M} \sum_{i:\sigma_i=1} \sum_{j:t_i<t_j} \mathcal{I}\left[\hat{y}_i\left(x_i\right) > \hat{y}_j\left(x_j\right)\right], \tag{2}$$

where M is the number of comparable pairs and $\mathcal{I}[\cdot]$ is the indicator function. It ranges from 0.0 to 1.0. The larger the C-Index is, the better the model performance could be.

2.2. Deep Learning Methods for WSIs

There are many deep learning methods for gigapixel WSIs since this field has developed considerably over the past few decades. Here we focus on the most relevant, important, and representative methods among them. In this section, we will divide them into two categories, methods with and without graphs, for literature review. A summary of them is shown in Table 1.

2.2.1. Basics of Graph Convolution Networks

Given the graph adjacency matrix $\mathbf{A}$, the graph node embedding $\mathbf{X}$ is updated by the message passing function defined in $\mathbf{GCN}(\cdot)$ [29]. It is written as

$$\begin{aligned} \mathbf{GCN}(\mathbf{X} \mid \mathbf{A}) &= \hat{\mathbf{L}}\, \mathbf{X}\, \mathbf{W}, \\ \hat{\mathbf{L}} &= \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}}, \end{aligned} \tag{3}$$

where $\hat{\mathbf{L}}$ is the normalized graph Laplacian matrix, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is for self-loops, $\hat{\mathbf{D}} = \sum_j \mathbf{A}_{i,j}$ is a degree matrix, and $\mathbf{W}$ is a learnable matrix of GCN layer.

2.2.2. Methods with Graphs

In graph-based WSI analysis, embedding-level methods often take patches as graph nodes and then connect some patches to manifest a graph. DeepGraphSurv [19] is the first work that introduces graphs into WSIs for survival prediction. It learns the Mahalanobis distances between all patches and then inputs them into a prior Gaussian model to compute edge connectivity, producing fully-connected graphs. Based on these graphs, spectral graph convolution is adopted to optimize patch embeddings. In DeepGraphSurv, the dimension of patch features is largely reduced by PCA to make its whole network feasible for computation. Afterward, RankSurv [20] and PatchGCN [21] are proposed one by one. The former builds hyper-graphs and the latter builds context-aware graphs. But the graphs in all of them are pre-fixed and cannot be updated during model training. The other kind of graph-based method is the cell graph network [10] that takes the other histopathological entities (*e.g.*, tumor cells) as graph nodes. However, they are not prevalent in WSI survival analysis due to their complex preprocessing procedures and prohibitive computational costs.

2.2.3. Methods without Graphs

For those methods without graphs, some of them usually generate patch clusters and then learn cluster embeddings. WSISA [13] is the first deep-learning method for WSI survival prediction. It trains a CNN for each cluster in which patches are labeled according to the label of its corresponding WSI, and then aggregates the outputs from CNNs. Based on WSISA, DeepAttnMISL [17, 18] uses an attention-based technique to weight different clusters dynamically, whereas BDOCOX [14] considers survival ranking for WSI prediction. After them, Constrained-MIL [25] is further proposed for ulcerative colitis prediction. It follows an instance-independency assumption and uses class-specific activation maps to extract the most significant features, which cannot be adapted to survival tasks. In addition, many self-attention-based methods [23, 30, 22, 24] are also proposed for WSI classification and demonstrate remarkable performances. However, the computation complexity of

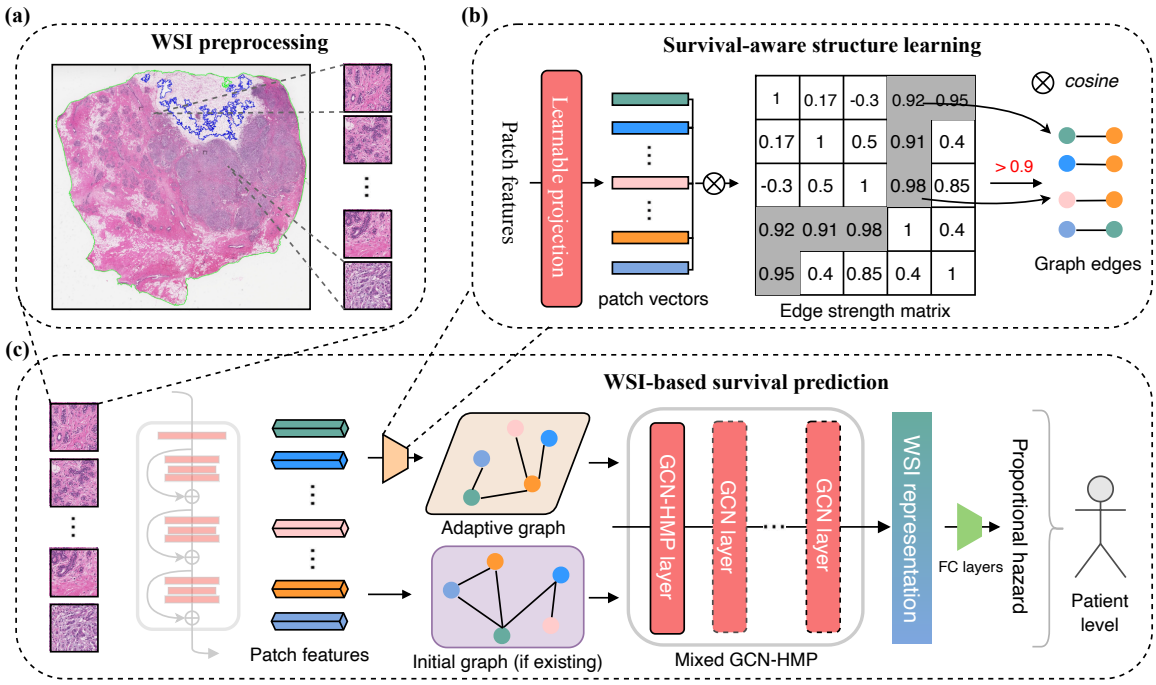

Figure 2: GraphLSurv: a graph-based weakly-supervised framework for survival prediction on WSIs. (a) WSI preprocessing that segments tissue region and outputs image patches of 256×256 pixels (approximately 0.5 microns per pixel). (b) Survival-aware structure learning that takes patch features as input and learns to build an adaptive and sparse structure. (c) Framework overview: patch feature extraction, graph construction, WSI representation learning, and proportional hazard estimation. GCN-HMP refers to a graph convolution network with hybrid message passing, which is elaborated in Section 3.2.2. FC indicates fully-connected.

vanilla self-attention [31] is quadratic, which could be memory-infeasible for extremely-large bags (*e.g.*, the bag with 10,000 patches). To tackle the challenge, HIPT [26] adopts patch hierarchy to reduce the number of tokens fed into Transformer [31]. This way requires a Transformer-based hierarchical pretraining for WSIs. Such a pretraining strategy is the focus of HIPT, which has large differences from our topic of graph-based representation learning. This paper considers more general scenarios in WSI survival analysis.

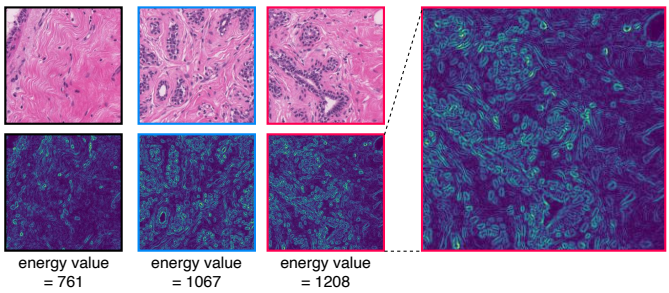

Figure 3: Example of the patch energy maps calculated with Sobel filter.

3. Methodology

As shown in Figure 2, our methodology mainly consists of three parts: WSI preprocessing, survival-aware structure learning, and GCN-based survival prediction. In this section, we will elaborate on each of them.

3.1. WSI Preprocessing

As shown in Figure 4, tissue regions are firstly segmented from WSIs. To further focus on key regions, the blank holes in segmented tissue regions are detected and discarded. Then, we slice these tissue regions into small tiles, yielding non-overlapping patches (whose number is often around 10,000) with 256×256 pixels.

It's extremely time-consuming and not necessary to directly train on all image patches. Thus, we design a sampling strategy to filter the image patches without obvious texture. Specifically, the energy map is firstly calculated for each patch using an efficient implementation of Sobel filter [32]. As shown in Figure 3, the patch image with obvious texture tends to have a large energy value. Based on these maps, then the top s patches with higher energy values are selected. In addition, following common preprocessing procedures, we apply the classical Color Normalization [33] to all patches for reducing the color inconsistency across different patches. Afterward, we utilize the friendly tool, CLAM [6], to extract patch features. Namely, a ResNet-50 model [34], pre-trained on ImageNet [35], is used as the feature extractor to transform each patch into a c-dimensional vector, where an adaptive average 2D-pooling layer that follows the third residual block of ResNet-50 is truncated for processing patches [6]. We specifically choose this ResNet-50 model as it is a strong baseline [26] and also has a reasonable computational cost for patch feature extracting [6].

After preprocessing, we denote the data of the i-th patient by $B_i = \left\{I_j \in \mathbb{R}^c\right\}_{j=1}^{s}$, where I_j is the j-th patch (or in-

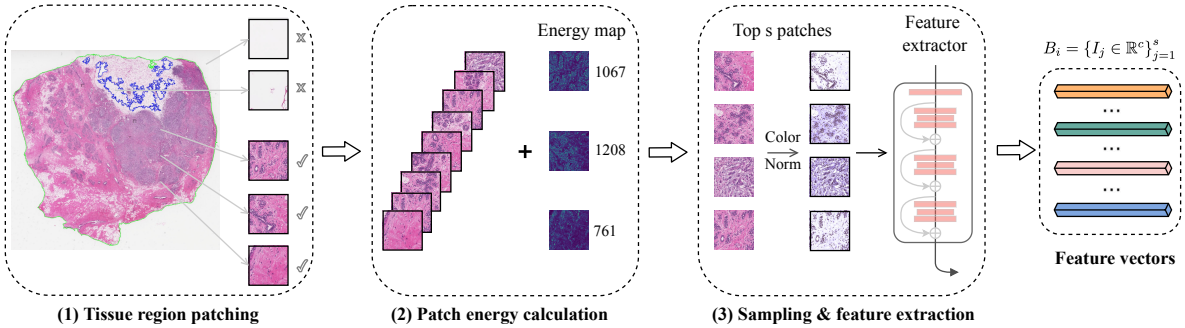

Figure 4: WSI preprocessing. (1) Tissue region patching: tissue region is segmented from WSI and then patches are uniformly drawn from tissue region. (2) Patch energy calculation: energy map and value are calculated using an efficient image filter, detailed in Figure 3. (3) Sampling and feature extracting: top s patches are sampled and normalized in color space, and their features are output from a feature extractor. B_i denotes a collection of feature vectors. I_j is a patch embedding.

stance) embedding. Note that the value of s may be different across patients (or bags). The complete dataset is denoted as $\{(B_i, t_i, \sigma_i)\}_{i=1}^n$. our task is to train a model F that predicts patients' survival hazard from their WSIs. The survival prediction of i-th patient is written as

$$\hat{y}_i\left(B_i\right) = \mathrm{F}\left(B_i; \theta\right),$$

where θ is the parameter of F.

3.2. WSI representation learning

With the absence of patch-level labels, we utilize a weakly-supervised approach to learn WSI representation from local unlabeled patches. Motivated by the fact that cancerous tissue regions could largely reflect tumor evolution and disease progression, we seek to enable the network to automatically learn an adaptive patch structure that could present some survival-related regions. Based on this patch structure, informative non-local embeddings can be desirably learned through patch feature aggregation, thereby obtaining effective representations for survival prediction.

3.2.1. Survival-Aware Structure Learning

A straightforward approach to building a patch structure is to connect each patch with its k nearest neighbors (k-NN) or adjacencies. While this approach is widely adopted in [19, 20, 21], we argue that such static structure may be sub-optimal. And most importantly, this structure often needs repetitive fine-tuning by manually changing k according to downstream tasks.

Thereby, we propose a structure learning strategy to generate adaptive and sparse structures, as illustrated in Figure 2(b). With a patch similarity learning technique that could optimally connect patches, some sub-structures in those generated structures are expected to properly define compact and informative regions. Note that our strategy could directly compute edge connectivity, not relying on a prior computation model adopted in DeepGraphSurv [19].

As described in Algorithm 1, the feature matrix $\mathbf{X} \in \mathbb{R}^{s\times c}$ of each bag is projected into a new feature space by a learnable projection matrix $\mathbf{T} \in \mathbb{R}^{c\times p}$. The new feature matrix is denoted by $\mathbf{P} \in \mathbb{R}^{s\times p}$. Then, cosine distance is calculated on each patch pair to measure patch similarities. This outputs a symmetric matrix $\mathbf{A}_L \in \mathbb{R}^{s\times s}$, where the magnitude of each element indicates edge connection strength. After that, we mask the element of $\mathbf{A}_L$ as 0 if its connection strength is less than a threshold δ. This operation produces a sparse structure that could filter unnecessary connections, thus decreasing noisy correlations in $\mathbf{A}_L$. We denote Algorithm 1 by a function $\mathcal{M}(\cdot)$:

$$\mathcal{M}(\mathbf{X}; \mathbf{T}, \delta) : \mathbb{R}^{s\times c} \rightarrow \mathbb{R}^{s\times s}. \tag{4}$$

The linear projection with learnable parameters is used to convert a raw feature space to a new one that is specially used for generating adaptive structures. Given a survival prediction task, these adaptive structures could become survival-aware as the network is optimized. We will investigate the advantages of adaptive and sparse structures in survival prediction through experiments.

It's worth noting that the proposed structure learning strat-

Algorithm 1: Survival-aware structure learning.

Input: Feature matrices of patients: $\{\mathbf{X}\}_{i=1}^n$; Learnable projection matrix: $\mathbf{T}$; Threshold: δ
Output: Adaptive and sparse graphs: $\{\mathbf{A}_L\}_{i=1}^n$

$\mathcal{R} \leftarrow$ set {}
foreach $\mathbf{X}$ *in* $\{\mathbf{X}\}_{i=1}^n$ **do**
  $s \leftarrow$ the number of rows in $\mathbf{X}$
  $\mathbf{P} \leftarrow \mathbf{X}\mathbf{T}$
  Initialize $\mathbf{A}_L$ randomly
  for $j \leftarrow 1$ **to** s **do**
    $\mathbf{P}(j) \leftarrow$ the j-th row vector of $\mathbf{P}$
    for $k \leftarrow 1$ **to** s **do**
      $\mathbf{P}(k) \leftarrow$ the k-th row vector of $\mathbf{P}$
      $\mathbf{A}_L(j,k) = \frac{\mathbf{P}(j)\cdot\mathbf{P}(k)}{\|\mathbf{P}(j)\|\|\mathbf{P}(k)\|}$
      if $\mathbf{A}_L(j,k) < \delta$ **then**
        $\mathbf{A}_L(j,k) \leftarrow 0$
      end
    end
  end
  Append $\mathbf{A}_L$ to set $\mathcal{R}$
end
return $\mathcal{R}$

egy has close connections with self-attention mechanisms [31]. We note that their key differences lie in two aspects: (1) our strategy aims to generate a graph structure as the necessary part to perform graph convolutions and beyond, whereas attention mechanisms are more general that focus on selecting critical information; (2) our graph structure is sparse while attention scores are densely distributed, which will be discussed in Section 4.5.2. Additionally, it is believed that such close connections are natural since there are inherent relations between GCN and self-attention, as demonstrated in [36].

3.2.2. Patch Embedding Learning and WSI Representation Deriving

We utilize graph convolution to update patch features as it enables a non-local node embedding learning through graph-based feature aggregation [29]. Specifically, each patch receives the messages (*i.e.*, patch features) from its connected patches, thus producing non-local patch features. The initial k-NN graph $\mathbf{A}_I \in \mathbb{R}^{s\times s}$ (if existing), built upon a raw feature space, may also have a contribution to survival prediction, so we combine it with $\mathbf{A}_L$ as a hybrid graph and perform *hybrid message passing* (HMP) on this graph. We denote this HMP by a function

$$\mathcal{F}(\mathbf{X} \mid \mathbf{A}_I, \mathbf{A}_L) : \mathbb{R}^{s\times c} \rightarrow \mathbb{R}^{s\times h}, \tag{5}$$

where h denotes output dimension. Specifically, $\mathcal{F}(\cdot)$ is implemented by

$$\begin{aligned} \mathbf{X}_I &\leftarrow \mathbf{GCN}(\mathbf{X} \mid \mathbf{A}_I), \\ \mathbf{X}_L &\leftarrow \mathbf{GCN}(\mathbf{X} \mid \mathbf{A}_L), \\ \mathcal{F}(\mathbf{X}) &= \lambda\mathbf{X}_I + (1-\lambda)\mathbf{X}_L, \end{aligned} \tag{6}$$

where λ is a hyper-parameter controlling the weight of $\mathbf{A}_I$.

Algorithm 2: Global WSI representation calculation.

Input: Feature matrices, initial graphs and adaptive graphs of bags: $\{\mathbf{X}, \mathbf{A}_I, \mathbf{A}_L\}_{i=1}^n$; Hyper-parameter: λ

Output: WSI representations: $\{\mathbf{S}_{rep}\}_{i=1}^n$

$\mathcal{R} \leftarrow$ set {}
foreach $\mathbf{X}$, $\mathbf{A}_I$, $\mathbf{A}_L$ *in* $\{\mathbf{X}, \mathbf{A}_I, \mathbf{A}_L\}_{i=1}^n$ **do**
  $\mathbf{E} \leftarrow \mathcal{F}(\mathbf{X} \mid \mathbf{A}_I, \mathbf{A}_L)$ using Equation (6)
  $h \leftarrow$ the number of columns in $\mathbf{E}$
  Initialize $\mathbf{S}_{\max} \in \mathbb{R}^h$ and $\mathbf{S}_{\text{avg}} \in \mathbb{R}^h$ randomly
  for $j \leftarrow 1$ **to** h **do**
    $\mathbf{S}_{\max}(j) \leftarrow \mathbf{max}_k \mathbf{E}(k, j)$
    $\mathbf{S}_{\text{avg}}(j) \leftarrow \mathbf{avg}_k \mathbf{E}(k, j)$
  end
  $\mathbf{S}_{\text{rep}} \leftarrow$ concatenate $\left(\mathbf{S}_{\max}, \mathbf{S}_{\text{avg}}\right)$
  Append $\mathbf{S}_{\text{rep}}$ to set $\mathcal{R}$
end
return $\mathcal{R}$

The derivation of WSI representation is shown in Algorithm 2. The feature matrix of each bag, $\mathbf{X}$, is updated by $\mathcal{F}(\cdot)$, outputting a new feature matrix $\mathbf{E} \in \mathbb{R}^{s\times h}$. The graph convolution layer applied to $\mathbf{A}_I$ and $\mathbf{A}_L$ is shared for saving memory overheads. Moreover, patch embeddings can be updated by multiple graph convolution layers with HMP. The WSI representation, $\mathbf{S}_{\text{rep}} \in \mathbb{R}^{2h}$, is output by concatenating the results from two commonly-used graph pooling operations, *i.e.*, max-pooling and avg-pooling.

3.2.3. Network Architecture

A standard GCN [29] performs graph convolutions only on a given graph. To make the GCN capable of structure learning and hybrid message passing, we propose a new GCN layer $\mathbf{GCN}_{\text{HMP}}(\cdot)$, named as GCN-HMP, implemented by

$$\mathbf{GCN}_{\text{HMP}}(\mathbf{X}, \mathbf{A}_I) = \mathcal{F}(\mathbf{X} \mid \mathbf{A}_I, \mathcal{M}(\mathbf{X})). \tag{7}$$

Based on this new layer, we present two architectures that will be used in GraphLSurv: pure GCN-HMP and mixed GCN-HMP (see Figure 5). After going through the first GCN-HMP layer, a hybrid graph will be generated in both pure GCN-HMP and mixed GCN-HMP. The biggest difference between these two architectures lies in: the structure of hybrid graph always remains unchanged as it goes deeper in mixed GCN-HMP, whereas in pure GCN-HMP it will be updated continuously by each layer. We will evaluate and clarify their functionality through experiments.

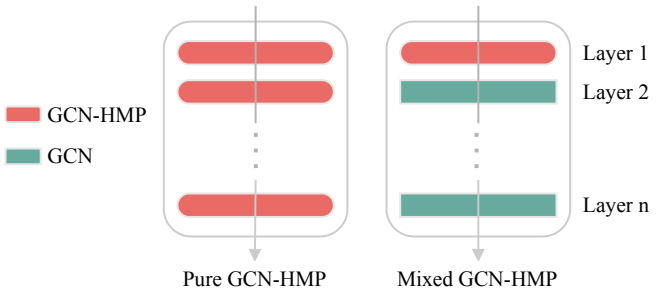

Figure 5: Network architectures of our GraphLSurv. A pure GCN-HMP is composed of GCN-HMP layers. A mixed GCN-HMP consists of one GCN-HMP layer and several standard GCN layers.

3.3. Survival Prediction

The derived WSI representation is utilized to calculate the proportional hazard $\hat{y}_i \in \mathbf{R}$ of i-th patient through fully-connected layers. To ensure the smoothness of the adaptive structure represented by a patch matrix $\mathbf{X}$ and an adjacency matrix $\mathbf{A}$, we add the Dirichlet energy [37] into our loss function:

$$\ell_{\text{graph}} = \frac{1}{s^2} tr(\mathbf{X}^T \mathbf{L} \mathbf{X}), \tag{8}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian, $\mathbf{D} = \sum_j \mathbf{A}_{i,j}$ is a degree matrix, and $tr(\cdot)$ is the trace of a matrix. This term enables patch embeddings to change smoothly across the patches connected by each other. Therefore, our final loss function is

$$\ell = \ell_{\text{cox}} + \alpha\, \ell_{\text{graph}}, \tag{9}$$

where $\alpha \in [0.0, 1.0]$ is a hyper-parameter.

3.4. Optimization for Large WSI Graphs

Algorithm 2 requires $O\left(s^2h\right)$ for time complexity and $O\left(s^2\right)$ for space complexity, which is reasonable for the bag containing around 1,000 patches. However, there are usually more than 1,000 or even up to 10,000 patches in a WSI. Thus, it is essential to optimize them for practical use, but previous graph-based methods rarely consider it.

To solve this problem, we absorb the works of scalable graph computation [38, 39], and adopt an anchor-based technique. Note that Chen et al. [39] proposed an approach of graph optimization and used a complex iterative scheme to adjust graph, which is based on contrast learning and aims to obtain a robust node embedding. By contrast, our approach is more concise that aims to capture the potential correlations between patches. Moreover, our basic idea is to present some compact and informative regions to extract effective WSI representations, which is different from [39].

3.4.1. Scalable GraphLSurv

In scalable GraphLSurv, given the bag B and its feature matrix $\mathbf{X} \in \mathbb{R}^{s\times c}$, we randomly sample r anchors from B ($r \ll s$). We denote the set of anchors as $R \subset B$ and its feature matrix as $\mathbf{Z} \in \mathbb{R}^{r\times c}$. After projecting $\mathbf{X}$ to $\mathbf{P}$ and $\mathbf{Z}$ to $\mathbf{Q}$, cosine distance is measured for the patches from B and the anchors from R, outputting a patch-anchor connectivity matrix $\mathbf{A}_{\text{pa}} \in \mathbb{R}^{s\times r}$ as follows:

$$\mathbf{A}_{\text{pa}}(j,k) = \frac{\mathbf{P}(j)\cdot\mathbf{Q}(k)}{||\mathbf{P}(j)||||\mathbf{Q}(k)||}. \tag{10}$$

Only the elements greater than δ in $\mathbf{A}_{\text{pa}}$ are retained while the others are set to 0.

The MP on the patch-anchor graph $\mathbf{A}_{\text{pa}}$ is modified as

$$\mathbf{X}_L = \mathbf{GCN}_{\text{pa}}(\mathbf{X}, \mathbf{A}_{\text{pa}}) = \widehat{\mathbf{A}}_{\text{pa}}\widetilde{\mathbf{A}}_{\text{pa}}^T\,\mathbf{X}\,\mathbf{W}, \tag{11}$$

where $\widehat{\mathbf{A}}_{\text{pa}} \in \mathbb{R}^{s\times r}$ is a column-normalized matrix, $\widetilde{\mathbf{A}}_{\text{pa}} \in \mathbb{R}^{s\times r}$ is a row-normalized matrix, and $\mathbf{X}_L \in \mathbb{R}^{s\times h}$ is a feature matrix updated by MP. Other than it, the calculations of $\mathbf{X}_I$, $\mathbf{E}$, and $\mathbf{S}_{\text{rep}}$ remain unchanged as before.

3.4.2. Complexity Analysis

Our scalable GraphLSurv reduces both time and space complexity via an anchor-based technique. Next, we will justify this technique according to our modifications.

Modification 1: cosine distance calculation. We only need to calculate the cosine distance between patches and anchors, and store the patch-anchor matrix, so the time and space complexity can be reduced from $O\left(s^2p\right)$ and $O\left(s^2\right)$ to $O(srp)$ and $O(sr)$, respectively.

Modification 2: patch-anchor message passing. We modify the original function $\mathbf{GCN}(\cdot)$ to $\mathbf{GCN}_{\text{pa}}(\cdot)$. The matrix multiplication in $\mathbf{GCN}(\cdot)$ defined by Equation (3), can be finished in time complexity $O\left(s^2h\right)$ via matrix right-associativity. The matrix multiplication of $\mathbf{GCN}_{\text{pa}}(\cdot)$ defined by Equation (11), only requires a time complexity $O(srh)$ via matrix right-associativity. Since we only store the patch-anchor matrix, the space complexity will be reduced from $O\left(s^2\right)$ to $O(sr)$.

Table 2: Statistic of dataset.

Level	Statistic	NLST	TCGA-BRCA
Overall	# Patients	449	978
	Death ratio (%)	35.9%	13.5%
	# WSIs	1,225	1,043
	# Patches	3,599,783	2,922,677
	# Sampled patches	1,145,836	971,737
Avg. on patients	# WSIs	2.7	1.1
	# patches	8,017.3	2,988.4
	# sampled patches	2,552.0	993.6
Max. on patients	# sampled patches	**6,000**	4,000

This anchor-based optimization makes the complexity of GraphLSurv decrease almost by a factor of s, both in time and space. Therefore, it is possible and reasonable for GraphLSurv to be applied in the real-world even if there are more than 10,000 patches in a WSI, as long as we set a proper sampling ratio for patch anchors.

4. Experiment

To validate the proposed GraphLSurv, we conduct extensive experiments including model performance evaluation, model risk grouping, preprocessing analysis, ablation study, and patch structure visualization. This section will show their results and present some notable findings in GraphLSurv.

4.1. Experimental Setting

4.1.1. Dataset

The first dataset is from The National Lung Screening Trial (NLST) for lung cancer research [40]. The second dataset is from The Cancer Genome Atlas (TCGA) for breast cancer research [41], called TCGA-BRCA. A dataset statistic is shown in Table 2. There are 1,225 slides and 1,043 slides in NLST and TCGA-BRCA, respectively. These two datasets are selected for evaluation since they have larger numbers of slides among several public WSI datasets. The maximum number of sampled patches is 6,000 in NLST, which indicates that it is necessary for us to develop an efficient framework to process them. In patch sampling, we set s to 1,000, which follows [20].

4.1.2. Model Comparison and Performance Evaluation

We mainly compare our method with the existing methods oriented to WSI survival prediction (see Table 1). Note that the experiments of self-attention models (see Table 1) cannot run successfully because of their prohibitive memory overheads (especially when the number of patches is more than 5,000). RankSurv [20] is not among our baselines because its code is not publicly-available. And we ignore the previous work [17] but turn to adopt DeepAttnMISL [18] for comparisons because DeepAttnMISL is an improved version of [17]. Constrained-MIL [25] is not adopted for comparisons as it is not adaptive to survival prediction.

Table 3: Performance comparison between GraphLSurv and other baselines.

Method	NLST		TCGA-BRCA	
	C-Index	Time-dependent AUC	C-Index	Time-dependent AUC
DeepConvSurv [13]	0.51578 ± 0.00757	-	0.53103 ± 0.03278	-
WSISA [13]	0.56257 ± 0.04512	0.55262 ± 0.03467	0.41659 ± 0.03440	0.40000 ± 0.03679
PatchGCN [21]	0.56292 ± 0.04959	0.57242 ± 0.05068	0.55874 ± 0.03967	**0.58352 ± 0.03621**
DeepGraphSurv [19]	0.63718 ± 0.03501	0.61125 ± 0.06111	0.53509 ± 0.06450	0.47982 ± 0.09333
DeepAttnMISL [18]	0.61872 ± 0.05475	0.62512 ± 0.07922	0.65886 ± 0.03877	0.52390 ± 0.05385
BDOCOX [14]	0.63627 ± 0.05443	0.63514 ± 0.07053	0.66096 ± 0.03614	0.54977 ± 0.04123
GraphLSurv (ours)	**0.66132 ± 0.05452**	**0.65197 ± 0.06685**	**0.68348 ± 0.05116**	0.57686 ± 0.05655

*The best ones are marked in **bold**

As DeepAttnMISL [18] did, the time-dependent AUC that measures the goodness of time-specific survival probability prediction is also added to evaluate models. Note that time-dependent AUC is not available for DeepConvSurv as it only makes patch-level predictions. 5-Year AUC is reported on NLST. 10-Year AUC is adopted for TCGA-BRCA due to its very long survival time.

4.1.3. Implementation details

We randomly split 80% data into training set and 20% data into test set, and retain 20% training set as validation set. The data splitting is conducted at patient-level and is stratified by the ratio of censored data, following WSISA [13]. We conduct each experiment by using 5-fold cross-validation and report the mean and standard deviation.

For model training, we run 300 epochs on training data with a batch size of 16, an initial learning rate of 0.0001, and an optimizer of Adam. The learning rate decays by a factor of 0.8 if validation loss no longer decreases for 5 epochs. The model that has a minimum validation loss during training is used for testing. For model hyper-parameters, we empirically set c = 1,024, $p = 128$, and $h = 128$. The other hyper-parameters, such as λ, δ, α, r, and model architecture, are tuned on the validation set and set as follows: for NLST, $k = 10$, $\lambda = 0.0$, $\delta = 0.8$, $r = 0.2s$, $\alpha = 0.3$, and the architecture is GCN-HMP×1; for TCGA-BRCA, $k = 10$, $\lambda = 0.3$, $\delta = 0.9$, $r = 0.1s$, $\alpha = 0.1$, and the architecture is GCN-HMP×1 + GCN×2. All experiments run on a machine with an Nvidia RTX A4000 GPU (12G) and an i5-8400 CPU. More details can be found in the source code available at *https://github.com/liupei101/GraphLSurv*.

4.2. Model Performance

The results of C-Index and time-dependent AUC are exhibited in Table 3. From Table 3, we could see two obvious results: (1) the C-Index of GraphLSurv outperforms previous methods in WSI survival prediction, with a 3.79% improvement over DeepGraphSurv on NLST and a 3.41% improvement over BDOCOX on TCGA-BRCA; (2) the time-dependent AUC of GraphLSurv surpasses all baselines on NLST and only falls behind PatchGCN on TCGA-BRCA. The first result shows the superiority of our method in survival risk discrimination. And the second result suggests that GraphLSurv is competitive in predicting long-term survival probability. We note that the 10-Year AUC of GraphLSurv is less than that of PatchGCN, with a drop of 1.14%. It is possibly due to that PatchGCN is trained with time-discrete labels. Moreover, these labels are normalized into [0, 1] so that their values cannot be influenced by the length of prediction years. However, PatchGCN tends to perform moderately in risk discrimination, as indicated by C-Index.

In addition, we have two other findings from Table 3. (1) For those clustering-based methods, DeepConvSurv and WSISA are the worst ones. And there are large gaps between the performance of them and others, which indicates that the models relying on explicit local labels could not perform better than embedding-level weakly-supervised methods. However, while BDOCOX is also a model based on local labels, it is surprisingly competitive over all previous methods, surpassing its counterpart DeepAttnMISL both in C-Index and AUC on two datasets. It may be largely caused by the survival ranking term adopted in BDOCOX. (2) For previous graph-based methods, DeepGraphSurv is the best one with a mean C-Index of 0.63718 on NLST, but on TCGA-BRCA it suffers a cliff-like drop. There are two possible reasons for this unstable performance: (a) the dense connections used in DeepGraphSurv could incur unnecessary patches in message passing; (b) the adoption of PCA dimension reduction could damage the original feature space of patches. We note that PCA is much necessary for DeepGraphSurv since its absence would bring prohibitive memory overheads to DeepGraphSurv. By contrast, our graph-based method instead preserves the original feature space and adopts an anchor-based technique to save the computational overheads. Moreover, GraphLSurv learns a sparse structure for graph convolution and avoids aggregating unnecessary information. We will conduct extensive experiments to understand what makes GraphLSurv perform better than others.

4.3. Prognostic Risk Grouping

The model ability of prognostic risk grouping is also a common criterion in survival analysis. As did in previous works [18, 14, 21], we classify patients into a high-risk prognostic group (short-term survivors) if their hazard predictions are larger than a cutoff value, otherwise, classify them into a low-risk one (long-term survivors). This cutoff value is derived from the median of hazard predictions. The method with the largest C-Index among all previous methods is selected to compare with GraphLSurv, which yields DeepGraphSurv and BDOCOX

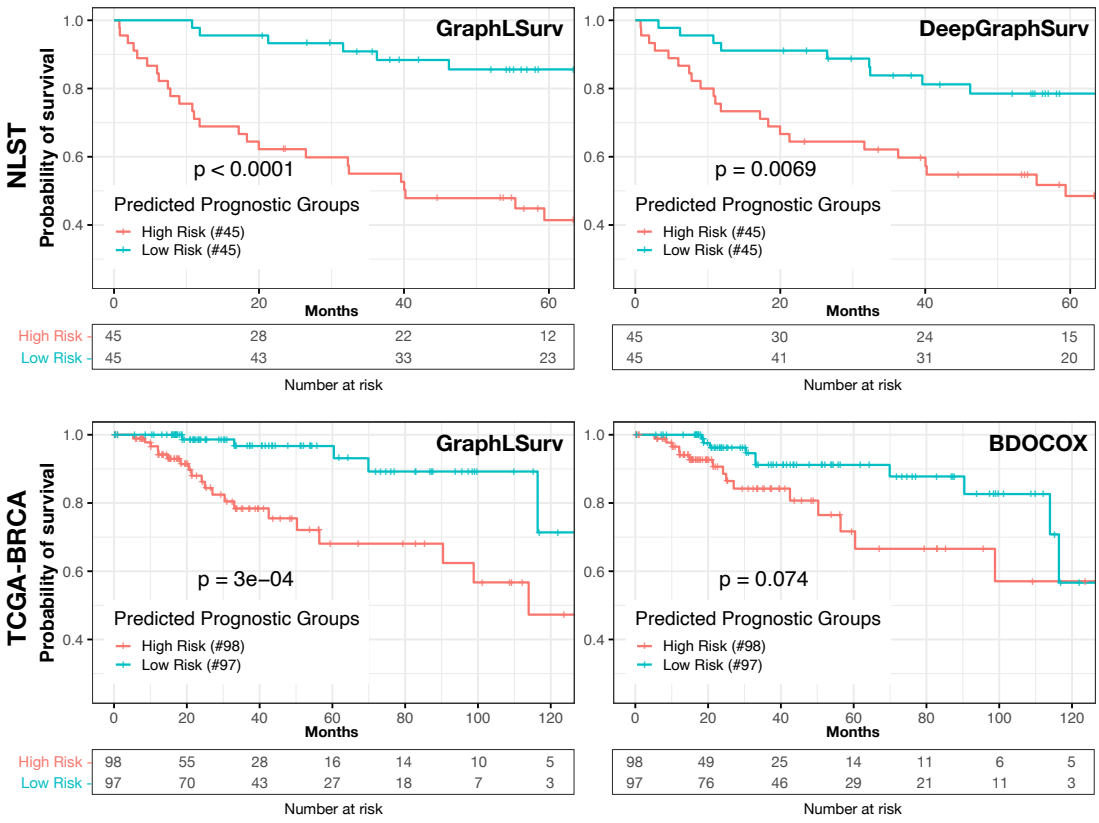

Figure 6: Kaplan-Meier curves of patient stratification. Each curve gives the ground truth survival of high- or low- risk patients. The actual number of survivors in each risk group is exhibited below survival curves. The P-value computed by log-rank test measures the survival difference between two groups.

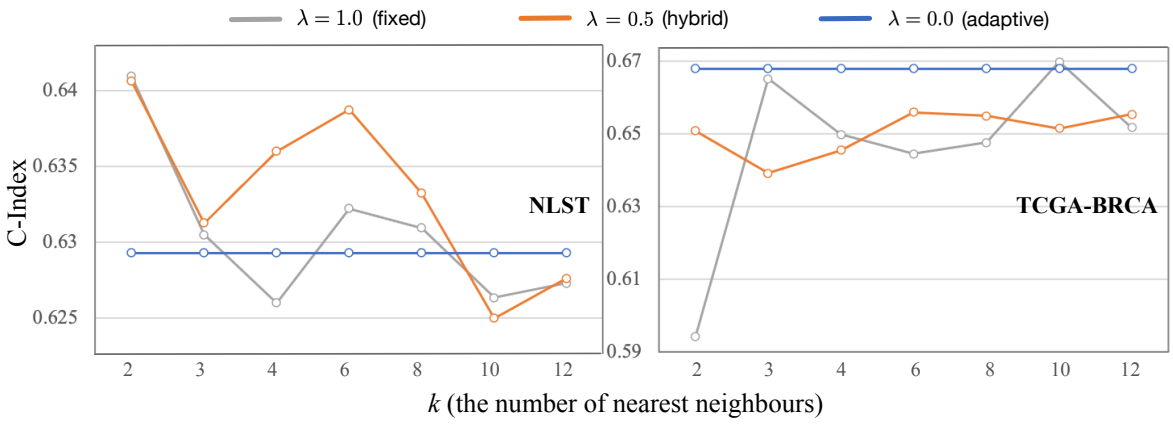

Figure 7: Ablation study to adaptive patch structure.

on NLST and TCGA-BRCA for comparisons, respectively. P-values are calculated by the log-rank test, which assesses the survival difference between risk groups. Note that the patients evaluated in risk grouping are from test sets.

From Figure 6, we can obviously see that GraphLSurv is superior in distinguishing between high- and low-risk patients, on both NLST and TCGA-BRCA. For example, on NLST, GraphLSurv recognizes high- and low-risk patients, with a P-value smaller than 0.0001 (VS. a P-value of 0.0069 given by DeepGraphSurv). For the results at the endpoint given by GraphLSurv, the number of survivors in high-risk group and low-risk group is 12 and 23, respectively. Whereas for DeepGraphSurv, 3 survivors are incorrectly predicted as high-risk patients. In addition, on TCGA-BRCA, our method also brings a significant difference with a P-value of 0.0003 (P-value $<$ 0.05), while BDOCOX fails to distinguish high- and low- survival risks significantly (P-value = 0.074).

4.4. Preprocessing Analysis

We conduct experiments on NLST to analyze the settings of WSI preprocessing (see Figure 4, stain normalization and WSI patching), keeping the same settings of other model parameters. The experimental results are shown in Table 4. Note that the evaluating time of stain normalization is measured by using 1,000 patches of 256 $\times$ 256 pixels. Two common python packages, torchstain and staintools, are utilized for Macenko [33] and Vahadane [42], respectively. For WSI patching, average time, listed in the last column, are obtained by using all the WSI samples with various patch sizes.

Table 4: Analysis of different settings in stain normalization (Norm) and WSI patching.

Stage	Setting	C-Index	Time (s)
Stain Norm	Macenko [33]	0.66132 ± 0.05452	10
	Vahadane [42]	0.65250 ± 0.04371	227
WSI Patching	128 × 128	0.61887 ± 0.06673	3.76
	256 × 256	0.66132 ± 0.05452	1.54
	512 × 512	0.64386 ± 0.03782	1.40

4.4.1. Stain Normalization

From the results shown in Table 4, we can see that the stain normalization of the Vahadane method falls slightly behind that of the Macenko method by 0.00882 in downstream prediction. Moreover, the Macenko method could run faster than the Vahadane method significantly, caused by the lack of GPU support in Vahadane's implementation [1].

4.4.2. Patch Size

As shown in Table 4, we find that the patch size of 256 $\times$ 256 is more suitable than other sizes like 128 $\times$ 128 and 512 $\times$ 512. In addition, the size of 128 $\times$ 128 could lead to a worse performance. One possible reason is that such a small patch size would lead to a lot of tissue patches and make instance features concentrate more on local issue expression, which could weaken global structural representation for a WSI from instance features. On the other hand, the instance features extracted from patches of 512 $\times$ 512 pixels are more coarse-grained than that from smaller patches, which could lose local fine-grained information and degenerate model performance. By contrast, the patch size of 256 $\times$ 256 pixels could achieve a good balance between retaining fine-grained features and deriving effective global representations. In terms of time, the computation time would increase as the patch size becomes smaller, because a smaller patch size means a larger number of patches in dataset.

[1] Python package staintools (https://github.com/Peter554/StainTools)

4.5. Ablation Study

The results given above have shown the power of GraphLSurv. Here we dive into the cores of GraphLSurv, *i.e.*, structure learning and hybrid message passing, to understand why our model works.

*4.5.1. **Adaptive structure is often more suitable than k-NN; k-NN structure requires fine-tuning according to tasks***

By adjusting the value of λ, we measure the performance of three different structures. We set the parameters related to structure learning by default: $\delta = 0.8$, $r = 0.1s$, and $\alpha = 0.1$. These three different structures are

- Fixed ($\lambda = 1.0$): graph convolution only performs on a k-NN structure.
- Hybrid ($\lambda = 0.5$): graph convolution performs on a hybrid structure containing a k-NN graph and an adaptive graph.
- Adaptive ($\lambda = 0.0$): graph convolution only performs on an adaptive structure.

From the results shown in Figure 7, we have two empirical findings: (1) the fixed k-NN graphs are not always suitable for survival prediction, which often requires fine-tuning according to running tasks; (2) the GCN with an adaptive structure (λ = 0.0/0.5) could often perform better than that with a fixed k-NN.

Our evidence has three-fold. (1) For $\lambda = 1.0$ shown by the gray line in Figure 7, its C-Index often changes with k, which ranges from 0.626 to 0.641 (0.63060 ± 0.00516) on NLST and from 0.594 to 0.670 (0.64612 ± 0.02465) on TCGA-BRCA. This means that some fixed k-NN structures may not work well on survival prediction. And an optimal k often requires repetitive fine-tuning. (2) If we don't involve k-NN graph in graph convolution at all, *i.e.*, $\lambda = 0.0$ shown by the blue line, our adaptive structure could achieve competitive results with a C-Index of 0.62928 on NLST, and outperform almost all results given by $\lambda = 1.0$ on TCGA-BRCA with a C-Index of 0.66786. It demonstrates the effectiveness of our adaptive structure. (3) If $\lambda = 0.5$, *i.e.*, taking both adaptive structure and k-NN graph into the model with a ratio of 0.5 (the orange line), we can find that the values of C-Index are improved compared to only using k-NN graphs on NLST, except for the k-NN graph with $k = 10$. It implies that some essential correlations missed in k-NN graphs may be successfully captured by structure learning. However,

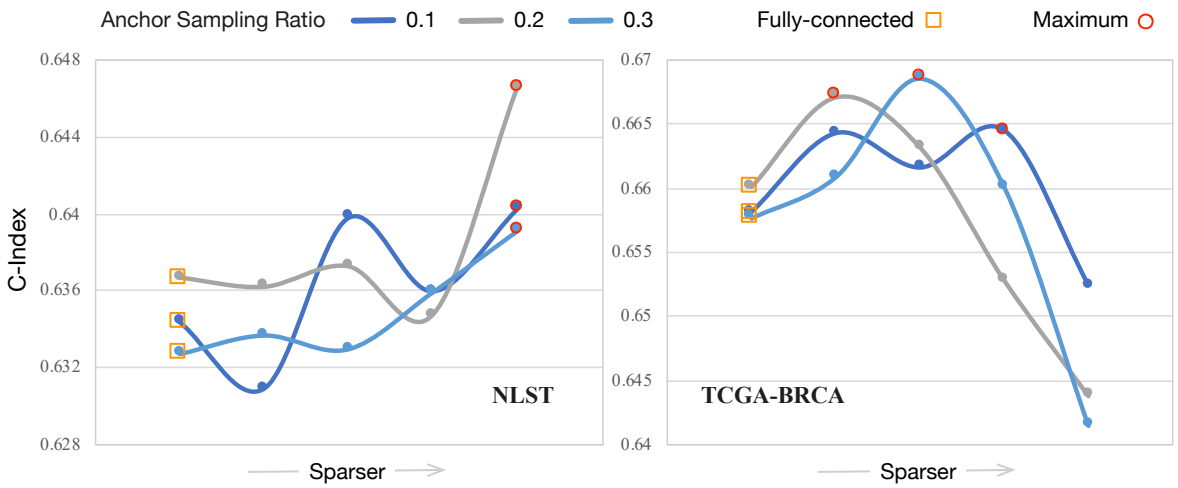

Figure 8: Ablation study to sparse patch structure.

it is not established on TCGA-BRCA when $k = 3$, 4, and 10, which means that adaptive structures may incur noises as well. This may require further noise filtering by setting a proper δ.

Note that adaptive structures are learned under default settings and we don't ever fine-tune them. The evidence above suggests that the graphs automatically learned by structure learning could be effective and superior both in performance and practicability, compared to those fixed k-NN graphs.

4.5.2. *Sparse structures are more likely to perform better than dense ones*

We examine the effects of structure sparsity on our model. The generated structures would be sparser if we set a larger δ. Two extreme cases could be taken into consideration: (1) $\delta = 0.0$: the connections between patches and anchors are built as a complete graph, similar to the style of self-attention mechanisms; (2) $\delta = 1.0$: there is no any connection between patches and anchors, which means that a patch embedding will be updated only by a message from itself and it makes no sense to evaluate this case.

Specifically, in this experiment, we set $\lambda = 0.0$ at first to preclude the potential effects of k-NN graphs on sparse structure learning. We adjust the value of δ from 0.0 to 0.8 with a step of 0.2. In addition, we set different ratios for anchor sampling to better understand the effect of sparse structures, because graph edges are connected between patches and anchors and the ratio of anchor sampling determines the upper bound of graph edge numbers. Moreover, the larger the anchor sampling ratio, the higher the maximum number of possible edges. Other hyperparameters are set by default.

As shown in Figure 8, we can empirically find that (1) the structure with high sparsity tends to perform better on NLST whereas the structure with moderate sparsity could often achieve a higher C-Index on TCGA-BRCA; (2) the dense structure in which patches and anchors are fully-connected like self-attention operations often leads to a sub-optimal model; (3) the structure with extreme high sparsity is not always the best.

Our observations have two-fold. (1) The best C-Index (red circle in Figure 8) is achieved by $\delta = 0.8$ on NLST, regardless of anchor sampling ratio. And on TCGA-BRCA, the best one is achieved by $\delta = 0.2/0.4/0.6$. However, the performances on fully-connected structures are often at a moderate or lower level. (2) On NLST, as the structure becomes sparser, the model tends to be better in C-Index. On TCGA-BRCA, the performance first increases and then decreases as the structure becomes sparser. Moreover, the maximum performance is often achieved at a moderate sparsity.

4.5.3. *Once structure learning may be adequate to capture essential patch correlations*

As described in Section 3.2.3, for mixed GCN-HMP, the model learn a survival-aware structure only at the first layer and this structure will be used for all remaining layers. But in pure GCN-HMP, the structure is learned and changed at any layers. Thus, we seek to study whether our structure learning should be used in the model only once or more times. Consequently, we set different numbers of layers both for a pure GCN-HMP and a mixed GCN-HMP, and then evaluate each of them.

Table 5: Model Architecture

Type	Architecture	C-Index	
		NLST	TCGA-BRCA
base	GCN-HMP×1	**0.66132**	0.67156
pure	GCN-HMP×2	0.65017	0.65913
	GCN-HMP×3	0.63846	0.66192
	GCN-HMP×4	0.63702	0.65854
	GCN-HMP×5	0.63454	0.65708
mixed	GCN-HMP×1 + GCN×1	0.64955	0.66363
	GCN-HMP×1 + GCN×2	0.63796	**0.68348**
	GCN-HMP×1 + GCN×3	0.63854	0.65544
	GCN-HMP×1 + GCN×4	0.63165	0.57817

As shown in Table 5, we can obviously see that a base model with GCN-HMP×1 achieves the best C-Index (0.66132) on

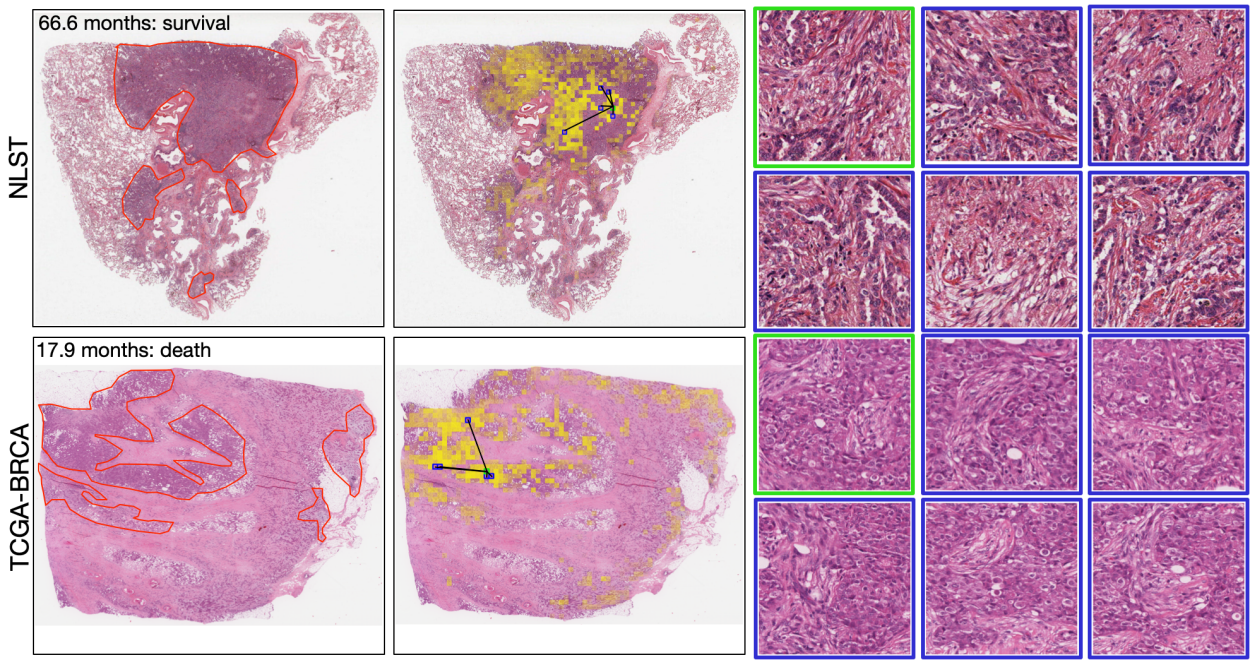

Figure 9: Visualization of some sub-structures learned by our structure learning method. The first column presents original WSIs. The red line indicates the annotation of region of interest. The second column gives the heatmaps that show the strength (indicated by yellow brightness) of the edges between a given patch and its connected patches. The other columns show the given patch (in the green rectangle) and its most closely-connected 5 patches (in the blue rectangle).

NLST and a mixed model with GCN-HMP×1 + GCN×2 obtains the best C-Index (0.68348) on TCGA-BRCA. It suggests that it would be better to use one GCN-HMP layer. We thereby infer that once structure learning could be adequate to capture essential correlations between patches for accurate prediction.

In addition, there are two other observations. (1) On both NLST and TCGA-BRCA, stacking more GCN-HMP layers would marginally harm the model performance of pure GCN-HMP. (2) On TCGA-BRCA, the model with 5 GCN-HMP layers still has a stable performance with a C-Index of 0.65708, while its counterpart (GCN-HMP×1 + GCN×4) suffers from a significant performance degeneration with only a C-Index of 0.57817. These empirical observations imply that the pure GCN-HMP model may be more stable than the mixed one stacked with standard GCN layers. However, a more comprehensive study is required to confirm this. We would leave it as our future work.

4.6. Patch Structure Visualization

We select two images to visualize the adaptive and sparse structures generated on them. Due to the tremendous patches, we select one representative patch from each slide for better visualization. From the result of structure visualization shown in Figure 9, we can see that (1) the regions presented by those closely-connected patch images could largely match the region of interest (ROI) in WSIs; (2) some noises, such as irrelevant patch images, also are included.

Specifically, for the WSI from the patient being right-censored at $t = 66.6$ months, its connected patches colored in deep-yellow could almost present a compact ROI completely, which could make our model much easier to learn meaningful non-local patch embeddings. However, there are still some missed patches for the given patch. In addition, the selected patch and its top-5 patches are very similar in color, morphology, and content. For the WSI from the patient who died at $t = 17.9$ months, those connected patches could almost cover the whole ROI, which could help our model aggregate those discriminative patches. But there are also some irrelevant patches involved in patch aggregating. These facts suggest that our structure learning method could automatically generate interpretable survival-aware structures; however, it still has an improvement space.

5. Discussion

Survival analysis nowadays plays an important role in disease understanding, patient management, and treatment decision-making. However, it is always a challenging task on whole-slide images as this kind of image has extremely-high resolution but only with weak labels. Although many MIL-based works have achieved SOTA results based on mainstream architectures (*e.g.*, Transformer and Attention-MIL), graph-based networks still have appealing merits in learning effective WSI representation as graphs are superior in expressing various sparse structures that could efficiently model the complex correlations in data [29, 36]. Our empirical results confirmed this, namely, the adaptive and sparse structures could often be more suitable for WSIs in survival prediction.

A limitation of GraphLSurv is the difficulty in learning structures with optimal sparsity, because the threshold δ for adjusting graph sparsity is fixed in training. In addition, there is an

improvement space for GraphLSurv in generating patch structure, as implied by our visualization results. Lastly, the study design of this work has some constraints, such as 1) a lack of comprehensive study on GCN-HMP features and advantages, which prevents us from better understanding the effect of GCN-HMP method, and 2) the limited dataset, which only covers two cancer types and cannot represent the rich diversity in the real world.

6. Conclusion and Future Work

In this paper, we developed a scalable survival prediction framework (GraphLSurv) with adaptive and sparse structure learning for histopathological WSIs. We mainly presented three cores of GraphLSurv: patch structure learning, GCN-HMP networks, and anchor-based optimization. We conducted extensive experiments on two publicly-available datasets to validate our GraphLSurv. Our comparative experiments demonstrated that GraphLSurv could often outperform previous models. Our further ablation study and empirical analysis suggested that the adaptive and sparse structures learned by GraphLSurv could be more suitable to describe WSIs for survival prediction. Moreover, only once graph structure learning could adequately capture essential correlations between WSI patches.

We hope that GraphLSurv could serve as an alternative and effective framework for WSI survival prediction. In the future, we will study a more robust and effective structure learning method to improve patch structures for WSI analysis.

References

[1] O.-J. Skrede, S. De Raedt, A. Kleppe, T. S. Hveem, K. Liestøl, J. Maddison, H. A. Askautrud, M. Pradhan, J. A. Nesheim, F. Albregtsen, I. N. Farstad, E. Domingo, D. N. Church, A. Nesbakken, N. A. Shepherd, I. Tomlinson, R. Kerr, M. Novelli, D. J. Kerr, H. E. Danielsen, Deep learning for prediction of colorectal cancer outcome: a discovery and validation study, The Lancet 395 (10221) (2020) 350–360. doi:10.1016/S0140-6736(19)32998-8.

[2] R. J. Chen, M. Y. Lu, D. F. Williamson, T. Y. Chen, J. Lipkova, Z. Noor, M. Shaban, M. Shady, M. Williams, B. Joo, F. Mahmood, Pan-cancer integrative histology-genomic analysis via multimodal deep learning, Cancer Cell 40 (8) (2022) 865–878.e6. doi:10.1016/j.ccell.2022.07.004.

[3] J. N. Kather, A. T. Pearson, N. Halama, D. Jäger, J. Krause, S. H. Loosen, A. Marx, P. Boor, F. Tacke, U. P. Neumann, H. I. Grabsch, T. Yoshikawa, H. Brenner, J. Chang-Claude, M. Hoffmeister, C. Trautwein, T. Luedde, Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer, Nature Medicine 25 (7) (2019) 1054–1056. doi:10.1038/s41591-019-0462-y.

[4] P. Tarantino, L. Mazzarella, A. Marra, D. Trapani, G. Curigliano, The evolving paradigm of biomarker actionability: Histology-agnosticism as a spectrum, rather than a binary quality, Cancer Treatment Reviews 94 (2021) 102169. doi:10.1016/j.ctrv.2021.102169.

[5] Y. Jiao, J. Li, C. Qian, S. Fei, Deep learning-based tumor microenvironment analysis in colon adenocarcinoma histopathological whole-slide images, Computer Methods and Programs in Biomedicine 204 (2021) 106047. doi:10.1016/j.cmpb.2021.106047.

[6] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, F. Mahmood, Data-efficient and weakly supervised computational pathology on whole-slide images, Nature Biomedical Engineering 5 (6) (2021) 555–570. doi:10.1038/s41551-020-00682-w.

[7] J. Lou, J. Xu, Y. Zhang, Y. Sun, A. Fang, J. Liu, L. A. J. Mur, B. Ji, PPsNet: An improved deep learning model for microsatellite instability high prediction in colorectal cancer from whole slide images, Computer Methods and Programs in Biomedicine 225 (2022) 107095. doi:10.1016/j.cmpb.2022.107095.

[8] W. Bulten, M. Balkenhol, J.-J. A. Belinga, A. Brilhante, A. Çakır, L. Egevad, M. Eklund, X. Farré, K. Geronatsiou, V. Molinié, G. Pereira, P. Roy, G. Saile, P. Salles, E. Schaafsma, J. Tschui, A.-M. Vos, B. Delahunt, H. Samaratunga, D. J. Grignon, A. J. Evans, D. M. Berney, C.-C. Pan, G. Kristiansen, J. G. Kench, J. Oxley, K. R. M. Leite, J. K. McKenney, P. A. Humphrey, S. W. Fine, T. Tsuzuki, M. Varma, M. Zhou, E. Comperat, D. G. Bostwick, K. A. Iczkowski, C. Magi-Galluzzi, J. R. Srigley, H. Takahashi, T. van der Kwast, H. van Boven, R. Vink, J. van der Laak, C. Hulsbergen-van der Kaa, G. Litjens, Artificial intelligence assistance significantly improves Gleason grading of prostate biopsies by pathologists, Modern Pathology 34 (3) (2021) 660–671. doi:10.1038/s41379-020-0640-y.

[9] C. Sun, B. Li, G. Wei, W. Qiu, D. Li, X. Li, X. Liu, W. Wei, S. Wang, Z. Liu, J. Tian, L. Liang, Deep learning with whole slide images can improve the prognostic risk stratification with stage III colorectal cancer, Computer Methods and Programs in Biomedicine 221 (2022) 106914. doi:10.1016/j.cmpb.2022.106914.

[10] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, L. Petersson, A survey on graph-based deep learning for computational histopathology, Computerized Medical Imaging and Graphics 95 (2022) 102027. arXiv:2107.00272, doi:10.1016/j.compmedimag.2021.102027.

[11] M.-A. Carbonneau, V. Cheplygina, E. Granger, G. Gagnon, Multiple instance learning: A survey of problem characteristics and applications, Pattern Recognition 77 (2018) 329–353. doi:10.1016/j.patcog.2017.10.009.

[12] K.-H. Yu, C. Zhang, G. J. Berry, R. B. Altman, C. Ré, D. L. Rubin, M. Snyder, Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features, Nature Communications 7 (1) (2016) 12474. doi:10.1038/ncomms12474.

[13] X. Zhu, J. Yao, F. Zhu, J. Huang, WSISA: Making Survival Prediction from Whole Slide Histopathological Images, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 6855–6863. doi:10.1109/CVPR.2017.725.

[14] W. Shao, T. Wang, Z. Huang, Z. Han, J. Zhang, K. Huang, Weakly Supervised Deep Ordinal Cox Model for Survival Prediction From Whole-Slide Pathological Images, IEEE Transactions on Medical Imaging 40 (12) (2021) 3739–3747. doi:10.1109/TMI.2021.3097319.

[15] L. Qu, X. Luo, M. Wang, Z. Song, Bi-directional weakly supervised knowledge distillation for whole slide image classification, Advances in Neural Information Processing Systems (2022).

[16] X. Wang, Y. Yan, P. Tang, X. Bai, W. Liu, Revisiting multiple instance neural networks, Pattern Recognition 74 (2018) 15–24.

[17] J. Yao, X. Zhu, J. Huang, Deep multi-instance learning for survival prediction from whole slide images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 11764 LNCS, Springer, 2019, pp. 496–504. doi:10.1007/978-3-030-32239-7_55.

[18] J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, J. Huang, Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks, Medical Image Analysis 65 (2020) 101789. doi:10.1016/j.media.2020.101789.

[19] R. Li, J. Yao, X. Zhu, Y. Li, J. Huang, Graph CNN for survival analysis on whole slide pathological images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 11071 LNCS, Springer, 2018, pp. 174–182. doi:10.1007/978-3-030-00934-2_20.

[20] D. Di, S. Li, J. Zhang, Y. Gao, Ranking-Based Survival Prediction on Histopathological Whole-Slide Images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 12265 LNCS, Springer, 2020, pp. 428–438. doi:10.1007/978-3-030-59722-1_41.

[21] R. J. Chen, M. Y. Lu, M. Shaban, C. Chen, T. Y. Chen, D. F. Williamson, F. Mahmood, Whole Slide Images are 2D Point Clouds: Context-Aware Survival Prediction Using Patch-Based Graph Convolutional Networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 12908 LNCS, Springer, 2021, pp. 339–349. arXiv:2107.13048, doi:10.1007/978-3-030-87237-3_33.

[22] H. Li, F. Yang, Y. Zhao, X. Xing, J. Zhang, M. Gao, J. Huang, L. Wang, J. Yao, DT-MIL: Deformable Transformer for Multi-instance Learning

on Histopathological Image, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 12908 LNCS, Springer, 2021, pp. 206–216. doi:10.1007/978-3-030-87237-3_20.

[23] B. Li, Y. Li, K. W. Eliceiri, Dual-stream Multiple Instance Learning Network for Whole Slide Image Classification with Self-supervised Contrastive Learning, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 14313–14323. arXiv:2011.08939, doi:10.1109/CVPR46437.2021.01409.

[24] B. Fu, M. Zhang, J. He, Y. Cao, Y. Guo, R. Wang, StoHisNet: A hybrid multi-classification model with CNN and Transformer for gastric pathology images, Computer Methods and Programs in Biomedicine 221 (2022) 106924. doi:10.1016/j.cmpb.2022.106924.

[25] R. del Amor, P. Meseguer, T. L. Parigi, V. Villanacci, A. Colomer, L. Launet, A. Bazarova, G. E. Tontini, R. Bisschops, G. de Hertogh, J. G. Ferraz, M. Götz, X. Gui, B. Hayee, M. Lazarev, R. Panaccione, A. Parra-Blanco, P. Bhandari, L. Pastorelli, T. Rath, E. S. Røyset, M. Vieth, D. Zardo, E. Grisan, S. Ghosh, M. Iacucci, V. Naranjo, Constrained multiple instance learning for ulcerative colitis prediction using histological images, Computer Methods and Programs in Biomedicine 224 (2022) 107012. doi:10.1016/j.cmpb.2022.107012.

[26] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, F. Mahmood, Scaling vision transformers to gigapixel images via hierarchical self-supervised learning, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2022, pp. 16123–16134.

[27] D.R.Cox, Regression Models and Life-Tables, Journal of the Royal Statistical Society. Series B (Methodological) (1972). doi:10.2307/2985181.

[28] P. J. Heagerty, Y. Zheng, Survival model predictive accuracy and ROC curves, Biometrics 61 (1) (2005) 92–105. doi:10.1111/j.0006-341X.2005.030814.x.

[29] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 5th International Conference on Learning Representations, ICLR 2017 (sep 2017). arXiv:1609.02907.

[30] Z. Shao, H. Bian, Y. Chen, Y. Wang, J. Zhang, X. Ji, yongbing Zhang, TransMIL: Transformer based Correlated Multiple Instance Learning for Whole Slide Image Classification, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 2136–2147.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 2017-December (2017) 5999–6009. arXiv:1706.03762.

[32] S. Avidan, A. Shamir, Seam carving for content-aware image resizing, ACM Transactions on Graphics 26 (3) (2007) 10. doi:10.1145/1276377.1276390.

[33] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, Xiaojun Guan, C. Schmitt, N. E. Thomas, A method for normalizing histology slides for quantitative analysis, in: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, IEEE, 2009, pp. 1107–1110. doi:10.1109/ISBI.2009.5193250.

[34] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[36] S. Zhang, S. Yan, X. He, LatentGNN: Learning efficient non-local relations for visual recognition, in: K. Chaudhuri, R. Salakhutdinov (Eds.), 36th International Conference on Machine Learning, ICML 2019, Vol. 2019-June of Proceedings of Machine Learning Research, PMLR, 2019, pp. 12767–12776. arXiv:1905.11634.

[37] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, MIT Press, 2002. doi:10.7551/mitpress/1120.003.0080.

[38] L. Wu, I. E.-H. Yen, Z. Zhang, K. Xu, L. Zhao, X. Peng, Y. Xia, C. Aggarwal, Scalable Global Alignment Graph Kernel Using Random Features, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, New York, NY, USA, 2019, pp. 1418–1428. doi:10.1145/3292500.3330918.

[39] Y. Chen, L. Wu, M. J. Zaki, Iterative deep graph learning for graph neural networks: Better and robust node embeddings, Advances in Neural Information Processing Systems 2020-December (jun 2020). arXiv:2006.13009.

[40] N. L. S. T. R. Team, The National Lung Screening Trial: Overview and Study Design, Radiology 258 (1) (2011) 243–253. doi:10.1148/radiol.10091808.

[41] C. Kandoth, M. D. McLellan, F. Vandin, K. Ye, B. Niu, C. Lu, M. Xie, Q. Zhang, J. F. McMichael, M. A. Wyczalkowski, M. D. Leiserson, C. A. Miller, J. S. Welch, M. J. Walter, M. C. Wendl, T. J. Ley, R. K. Wilson, B. J. Raphael, L. Ding, Mutational landscape and significance across 12 major cancer types, Nature 502 (7471) (2013) 333–339. doi:10.1038/nature12634.

[42] A. Vahadane, T. Peng, A. Sethi, S. Albarqouni, L. Wang, M. Baust, K. Steiger, A. M. Schlitter, I. Esposito, N. Navab, Structure-preserving color normalization and sparse stain separation for histological images, IEEE Transactions on Medical Imaging 35 (2016) 1962–1971. doi:10.1109/TMI.2016.2529665.