DAS-GCN: Differentiable Arch Search for Graph Convolutional Networks on Whole Slide Images Based Survival Prediction

[1] **Bingjian Yao,** [2]**Zhengfeng Zhuang,** [3]**Yanfeng Li,** [4]**Yan He,** [5]**Qiong Peng**

[1] 23020221154136 [2] 23020221154183 [3] 23020221154097 [4] 23020221154088 [5] 23020221154173
Yanfeng Li from AI class, rest of all from Information Institute class

Abstract

The task of survival prediction based on clinical and followup data of cancer patients is challenging. Although the Graph Neural Network methods of Multi Instance Learning (MIL) has made great progress in this field by organizing the Whole Slide Images (WSIs) of patients, the effect of different methods varies with the context-aware informatioin of the type of cancer. In this paper, inspired by the differentiable search scheme of Pooling Architecture Search for graph classification, we present DAS-GCN, a survival prediction scheme on WSIs based on the proposed Differentiable Arch Search framework for finding a suitable pre-defined graph convolutional network. Our method consists of three major components. Firstly, we propose a differentiable framework for searching each module of a GNN. Furthermore, pretrained finetuning is used to improved the generalization of the searched model. Finally, based on the first two steps, we propose a context-aware WSI space construction network and used the multi-level Graph Readout method to predict the risk of survival analysis. Experiments will be conducted on six TCGA cancer datasets.

Introduction

Survival analysis of clinical survival data aims to analyze and help clinicians make early decisions about treatment, which is crucial for the patients' healthware. Howerver, censoring attribute of the survival data makes survival analysis different from other prediction approaches. Traditionally, the most popular model for survival analysis is Cox proportional hazards model [1]. However, the Cox model is based on linear assumption and fails to consider the time factor, which is too simply for real-world scenarios. Therefore, more survival models are needed to fit nonlinear scenarios.

The prosperity of deep learning especially GNNs drives reseachers to explore DL-based survival analysis models on graph structure data. Such GNNs (GCN [2], GAT [3], GIN [4] and GraphSAGE [5]), make it possible to obtain the topological information in the graph structure with its contextaware. For example, on survival analysis task, Li R et al. [6] proposed DeepGraphSurv, which use intermediate patch-wise features to construct graph structure, and use GCN to integrate random local patch features with global topological structures. Instead of sampling random patch features from WSIs as nodes and connecting these nodes on the embedding space, Patch-GCN [7] building graphs via adjacent patches and it performs better than DeepGraphConv.

Recently, Neural Architecture Search (NAS) methods were successfully transfered from automatically searching state of the art (SOTA) CNN architectures to SOTA GNN architectures in a pre-defined search space and representative methods, such as GraphNAS [8], Auto-GNN [9], AutoGM [10], etc. However, most of these methods search on a discrete space, which is to large. The recently proposed differentiable search framework accelerates the learning time by turning the discrete problem into a continuous problem, which can search the neural network architecture efficicently. Therefore, inspired by PAS(Pool Architecture Search) [11], we proposed DAS-GCN on WSIs for survival prediction. Our methods consists of three parts: a) A differentiable framework for automatically searching each module of a GNN. b) Generalization improvement of the searched model by pretrained finetuning. c) A context-aware WSI space construction network used for predicting the risk of survival analysis after the multi-level Graph Readout. And we will conduct experiements on six TCGA canner datasets. Our contributions can be as followed:

- Firstly, we propose a differentiable framework automatically searching each module of a GNN, which can narrow the search space and improve the efficiency.
- Furthermore, we adopt pretrained finetuning to improve the generalization of the searched model.
- Finally, we use a context-aware WSI space construction network to predict the risk of survival analysis after multi-level Graph Readout.

Related Work

Graph Neural Network

The birth of GNNs make it possible to process the data represented in non-Euclidean space, such as graph data. Thus, GNNs can be applied to meet the requirements of various graph related learning tasks.

In general, graph neural networks can be divided into five categories based on a classification proposed by Zonghan Wu et al. [12], including RecGNN (recurrent graph neural network), GCN (graph convolutional networks), AutoGNN

(graph autoencoder), STGCN (spatio-temporal graph convolutional network) and GAT (graph attention network).

RecGNN extended from RNN model and it updates node states by recursively exchanging domain information based on information diffusion mechanism. GCNs was based on two parallel model methods, spatial based method and spectral based method respectively. And it borns with great sucess of CNN in computer vision [13]. AutoGNN [14] was deep neural networks that map nodes to latent feature spaces and decode graphical information from their latent representations. STGCN [15] was generated to adapt to the dynamic nature of graph structure and graph input which can be effectively applied to sequence data. GAT [3] was the network that introduces attention mechanism into traditional GNNs which can extract the different influence features of neighboring points on nodes.

Neural Architecture Search

NAS (Neural Architecture Search) methods proposed to automatically find outperformed CNN architectures in a certain search space have been transferred to GNN, e.g. GraphNAS [8], AutoGNN [9], AutoGM [10], DSS [16], SANE [17], AutoGraph [18] and Policy-GNN [19]. However, most of the existing methods based on methods to select architectures from the discrete search space, which is computationally expensive. Recently proposed differentiable search algorithms construct an over-parameterized network and optimize this network with gradient descent due to the continuous relaxation of the search space. Representative work is DARTS [20] which approach the efficiency problem from a different angle. DARTS[20] continuously relax the search space instead of searching over a discrete set of candidate architectures, so that the architecture can be optimized with respect to its validation set performance by gradient descent. Compared with existing methods mentioned above, PAS [11] provides one search space that can cover existing pooling methods and one coarsening strategy to develop an efficient data-specific pooling architecture learning method. Therefore, we choose PAS to search for the outperformed architecture for graph-level tasks.

Survival Analysis

In recent years, based on the convenience of various data acquisition and advanced deep learning paradigms, many effective frameworks have been proposed for survival analysis in WSIs [6, 7, 21, 22]. For example, Zhu et al. [23] developed a WSISA manner by adaptively sampling and clustering patches from each WSI as the input to a convolutional neural network for survival prediction, which was the first trial of moving survival prediction onto WSIs. Li et al. [6] modeled each WSI as a graph and then proposed a graph convolutional neural network with attention learning that better serves the survival prediction by rendering the optimal graph representations of WSIs. Chen et al. [7] presented Patch-GCN, a context-aware, spatially-resolved patch-based graph convolutional network that hierarchically aggregates instance-level histology features to model local-level and global-level topological structures in the tumor micro-environment. However, due to the poor computation and iteration capability of neural networks, existing methods have limited efficiency in WSIs-based survival analysis.

Proposed Solution

In this section we will describe the proposed DAS-GCN, including Problem Formulation, Framework overview and the design and search space of predefined layer modules.

Problem Formulation

Assuming there are datasets with WSIs: $W = \{W_1, W_2, ..., W_N\}$ and related Clinical information $Y = \{(T_i, C_i)|i \in \{1, \cdots, N\}\}$, including survival time T and censorship status C. Multi Instance Learning (MIL) approach is widely used in survival analysis while Graph construction is the key step [7]. To construct graph G, firstly, it is necessary to screen the patches of WSI segmentation. First, the low-resolution WSI is converted from RGB to HSV color space, and then the foreground space mask of effective pathological tissue is obtained by the binary threshold discrimination method. Each slide at $20\times$ magnification is split into the number of small patches $P_i = \{p_{i,j}|i \in \{1, \cdots, N\}, j \in \{1, 2, \cdots, n_i\}\}$ with non-overlapping by PyHIST [24], n_i is the number of patches that the i^{th} slide is divided into. We define a feature extractor: $d : p_{i,j} \rightarrow f_{i,j}, f_{i,j} \in F_i$, so that each patch uses ImageNet pre-trained ResNet50 Network to extract 1024-dimensional features and serve as the node features of Graph. After the location information $(x_{i,j}, y_{i,j})$ coordinates saved by patch, we use to build an adjacency matrix A_i for each W_i via K Nearest Neighbor Algorithm (k = 8) that models a 3 $\times$3 image receptive field in CNN convolutions. Then define the constructed graph as $G < F_i, A_i >$ as the input to the network model. Survival prediction tasks usually employ the cox proportional-hazards model. Survival Function is denoted as $S(t)$ and represents the probability that the survival time of the observed object crosses the time point t. The risk function represents the survival time after reaching t. The probability of failure, we can use $h(t)$ to express, $h(t) = f(t)/S(t)$. Where f(t) is the probability density function, the probability density function is the reciprocal of the cumulative distribution function $F(t)$ (Cumulative Distribution Function), and the relationship between the cumulative distribution function and the survival function is $F(t) = 1-S(t)$. The cumulative distribution function represents the probability that survival time does not exceed time point t. Cumulative Hazard Function $H(t) = -logS(t)$ For the COX model, we have $h(t, x) = h_0(t)exp(\beta X)$, while X is relevant factors that may affect survival time are covariates that do not change over time, while is also the premise of the Cox regression model, and hazards h requires the model to predict.

Framework Overview

We propose a framework DAS-GCN to automatically learn data-specific architectures of Graph Convolutional Network, which consists of Aggregation, Pooling, Readout operation and Multi-Level Graph representation Merge Modules [11]. Based on this framework, an efficient search structure is proposed, where we embed convolution, pooling and readout

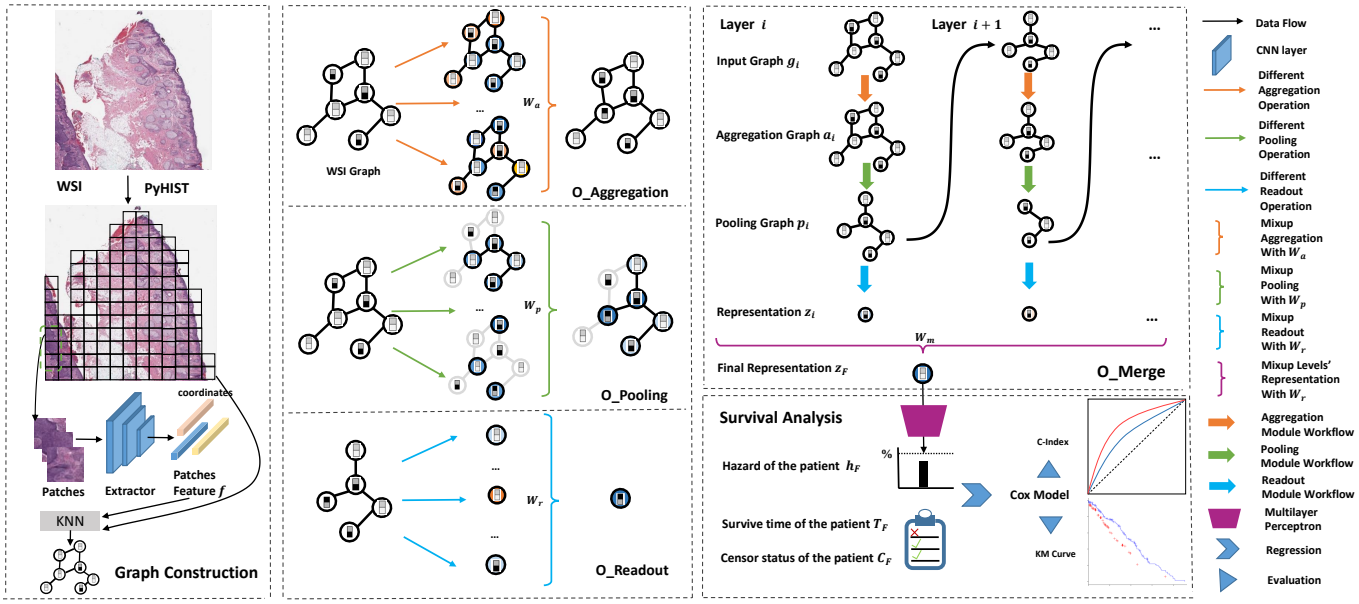

Figure 1: Overview of DAS-GCN

modules in each layer, and predefine a suitable search space in each module. Each module is described in detail below. As shown in Figure 1, for the input Graph G_i of the i^{th} layer, Aggregation Module aggregates first-order neighborhood features. By iteratively aggregating the representations of its k-num neighbors, the own representation $G_i^a =< A_i, F_i^a >$ capturing the structural information within k-hop neighborhoods can be generated. Then the graph will be passed to the Pooling Module to generate the graph $G_i^p =< A_{pi}, F_{pi}^a >$. After the Pooling Module, the Readout module is used for condensing representation of G_i^p to generate z_i of the i^{th} layer. Finally we merge hierarchically all pooling graphs to learn WSI representation features at different scales better result in z^F. After that MLP layers are used to generate hazards of the patients according to the Cox regression Model.

Table 1: Candidate operations in each module

Aggregation O_a	GCN, GAT, SAGE,GIN, TransformerConv, etc.
Pooling O_p	TopKPooling, SagPooling HopPoing, Global Attention Pooling, *NONE*, etc.
Readout O_r	Global_{sort}, Global_{att}, Global_{max}, Global_{mean}, etc.
Merge O_m	LSTM, MAX, Concat, MEAN, SUM, etc

Search Space of Predefined Layer Modules

Based on this framework, an effective search space can be designed naturally by including artificially designed operations, we design one search space with a set of candidate operations as shown in Table 1. We add *NONE* rations in the pool module, which means that pool operations are not used.

We assign learnable weight to the operation of each module in each layer $w_{i,M}, M \in \{a, p, r, m\}$, after learning the weight of the network structure, load the corresponding parameters, and select single path for finetune, which conforms to the efficient network search mode, reduces the network parameters, and speeds up the network fitting.

Differentiable Search Method

In this section, we will explain in detail how to search through the differentiable graph network, so that the relaxation operation for each layer can be carried out.

Generally speaking, for several predefined operations of each module, the relaxation operation and differentiable weighted summation can be performed by mapping the discrete space to the continuous space as follows:

$$O_M(x) = \sum_{i}^{|O_M|} W_{M,i} \bullet o_{M,i}(x)$$

where x represents the input representation of each module M, $W_{M,i}$ denotes the weights of operation of $o_{M,i}(\cdot)$. $W_{M,i}$ usually expressed as a reparameterization technique to determine the network structure parameters $W_{M,i}$, which is the corresponding learnable parameter for the input representation and the current state of the network.

Optimization and Loss function

The survival output of pathology modality output R_p can be calculated as :

$$R_p = MLP(z_F)$$

Therefore the Cox loss of R_p is calculated by:

$$L_{cox}^p = \sum_{i}^{B} \delta_i(-R_p(i) + log \sum_{j:t_j \geq t_i} exp(R_p(j)))$$

where B is the patients number of minibatch, $R_p(i)$ and $R_p(j)$ denote the survival output of finetune network of i^{th} and j^{th} patient. Regularization loss $L_{reg} = \|\theta\|$ is usually added to prevent the model from overfitting additionally. The structural parameters of W are already constrained at initialization, and regularization is unnecessary. In summary the total loss of the whole framework is define as:

$$L_{total} = \lambda_c L_{cox} + \lambda_{reg} L_{reg}$$

where λ_c and λ_{reg} are the trade-off parameters. The specific hyperparameter settings and implementation details can be found in Experiment part.

Experiments

To verify the effectiveness of the method proposed in this paper, in this section we will carry out comparison experiments and ablation experiments based on the datasets described below.

Datasets

Six of the largest cancer datasets from The Cancer Geno-me Atlas (TCGA), a public cancer data consortium that contains matched clinical records, diagnostic WSIs, and genomic data with labeled survival times and censorship statuses, were used to validate our proposed method.

We used the following cancer cohorts for this study: Glioma (GB-MLGG, 1011 cases), Kidney Renal Clear Cell Carcinom-a (KIRC, 385 cases), Liver Hepatocell-ular Carcinoma (LIHC, 287 cases), Lung Adenocarcinoma (LUAD, 452 cases), Lung Squamous Cell Carcinoma (LUSC, 438 cases), Uterine Corpus Endometrial Carcinoma (UCEC, 387 cases). The clinical records for each dataset are included in the supplementary material. All pathological slides are normalized to 10 magnification, with some patients having graph sizes as large as 30000 instances.

Evaluation Metric

As an evaluation metric, the concordance index (CI) is used. The CI index measures the proportion of all pairs of patients whose survival risks are correctly ordered. The confidence interval (CI) ranges from 0 to 1, with higher CI values indicating better survival prediction performance. The mean and standard deviation of 5 randomly repeated 5-fold CVs (5-fold CVs) are reported. It should be noted that during the cross-validation procedure, 20% of the training data is randomly split and tuned to determine the checkpoint.

Baselines

We use 4 different baseline methods, MIL method using summary of the instance set (MIL_Sum_FC), Attention-based deep multiple instance learning (ABMIL, MIL_Att_FC), Graph CNN for survival analysis on whole slide pathological images (DeepGraph) and Whold Slide Images are 2D Point Clouds: Context-Aware Survival Prediction using Patch-based GCN (Patch-GCN).

Among these methods, MIL_Sum_FC focuses on how to aggregate all of the instance information in WSI and it simply uses the sum operation to obtain the WSI-level information. MIL_Att_FC methods have received attention in computational pathology, where they have been used to solve problems such as cancer classification, cancer grading, and survival analysis to determine the most relevant instance of survival risk in a weighted case. As opposed to the conventional COX model, DeepSurv uses a deeper network (more than one hidden layer) and more sophisticated training methods like BatchNorm, Dropout, etc. Therefore, compared with the traditional cox model, DeepSurv does not need prior knowledge. Patch-GCN improves this advantage and highlights the relevance of generating graphs via neighboring patches rather than feature similarity in the embedding space. DeepGraph samples random patch characteristics from WSIs as nodes and connects these nodes on the embedding space. In contrast to cell-to-cell interactions between tumor cells and other cell types, DeepGraph has a higher c-Index on cancer kinds that correlates with global-level morphological variables including tumor size and depth of invasion in the myometrium.

Implementation details

We use ResNet50 pretrained on ImageNet Dataset to extract the features of instances with the size of 512 $\times$ 512. During training process, the dimension of each feature embedding is reduced from 1024 to 256 by a fully connected layer. Finally the feature embedding of each bag WSI can be represented as $H_i \in R^{n\times 512}$. We trained all the network with a batch size of 32 for 150 epochs for all datasets using Adaptive moment estimation (Adam) with initial learning rate of 7e-5 and weight decay of 1e-5 on NVIDIA GeForce RTX 3090Ti (24GB). The learning rate would be scheduled by Cosine learning rate scheduler with 1/4 period ending with 1e-7. In the inference step, the sigmoid is used to normalize the predicted scores for patients' hazards for all experiments.

Experiment Results

Performance Comparisons We used four methods with varying emphasis on six representative cancer datasets to evaluate the performance of the model compared with the methods presented in this paper.

As shown in Table 2, DAS-GCN has achieved the best performance on UCEC, LIHI, KIRC, LUAD and LUSC data sets, which indicates that the proposed method has excellent effect and better universality. Compared with MIL_Sum_FC method, our method is superior in all data sets, which indicates that our method can summarize instance sets better to form better global information. In addition, our method performs better than PatchGCN on all datasets, so it can be considered that our method can aggregate local information more effectively. Although the performance of our method on the larger dataset GBMLGG is slightly inferior to MIL_Attn_FC and DeepGraph, it still performs well on the remaining smaller datasets. This shows that our method can retain better global information even with a small amount of data, that is, our method has the advantage of not relying on data sets with a large amount of data for training. To sum up, our method can not only effectively retain global information and extract local information, but also has the

Table 2: Performance comparison between four SOTA methods with our proposed method on six of the largest cancer datasets from TCGA.

	DeepGraph	PatchGCN	MIL_Sum_FC_surv	MIL_Attn_FC_surv	**DAS-GCN (Ours)**
GBMLGG	0.7160 ±0.0134	0.7128 ±0.0157	0.5756 ±0.0051	**0.7181 ± 0.0183**	0.7153 ±0.0284
UCEC	0.6230 ±0.0378	0.6324 ±0.0114	0.5617 ±0.0167	0.6224 ±0.0291	**0.6460 ± 0.0271**
LIHC	0.6163 ±0.0197	0.6152 ±0.0202	0.5162 ±0.0061	0.6127 ±0.0128	**0.6259 ± 0.0195**
KIRC	0.6522 ±0.014	0.6657 ±0.0058	0.4831 ±0.02	0.6641 ±0.0101	**0.6883 ± 0.0153**
LUAD	0.5454 ±0.0326	0.5563 ±0.0135	0.4779 ±0.0245	0.5505 ±0.0215	**0.5734 ± 0.0246**
LUSC	0.5584 ±0.0197	0.5781 ±0.0192	0.4972 ±0.0041	0.5429 ±0.0173	**0.5791 ± 0.0149**

Table 3: Influence of Num_layers in search space.

	Num_Layer = 1	Num_Layer = 2	Num_Layer = 3	Auto_Search_Layer
GBMLGG	0.6947 ±0.0367	0.7057 ±0.0245	0.7116 ±0.0201	**0.7153 ± 0.0284**
UCEC	0.6247 ±0.021	0.6349 ±0.0281	0.6375 ±0.0279	**0.6460 ± 0.0271**
LIHC	0.6228 ±0.0401	0.6199 ±0.0379	0.6237 ±0.0255	**0.6259 ± 0.0195**
KIRC	0.6576 ±0.0235	0.6712 ±0.0244	0.6619 ±0.0214	**0.6883 ± 0.0153**
LUAD	0.5697 ±0.0171	**0.5824 ± 0.0211**	0.5701 ±0.0188	0.5734 ±0.0246
LUSC	0.5715 ±0.0245	0.5677 ±0.0116	0.5621 ±0.0222	**0.5791 ± 0.0149**

characteristics of not relying on large amounts of data, and the overall performance is quite excellent.

Ablation Studies An ablation study was performed to show the influences of the number of module layers in the search space, for which results are shown in Table 3. We search for the number of layers based on other fixed network architectures, and the cases where the layer number is 1, 2, 3, and automated are discussed. As shown in Table 3, the automated number achieved the best results on all datasets except LUAD. For example, compared to the fixed module layer numbers, the search strategy can improve accuracy with 1.7% and has the minimum standard variance in KIRC dataset. This observation demonstrates the importance of including the number of module layers in the search space, which can verify the superiority of the automated number searched by DAS-GCN over human-designed numbers.

Conclusion

Despite the progress made in the field of survival prediction, the different method varies with the context-aware information of the type of cancer. In this paper, we present DAS-GCN, a survival prediction scheme on WSIs based on the proposed Differentiable Arch Search framework for finding a suitable pre-defined graph convolutional network. In comparing DAS-GCN to the sort-of-the-art methods, we observe that DAS-GCN outperforms all prior approaches on 5 cancer types in the TCGA. Moreover, we demonstrate the improvement in selecting the numbers of network's layer by auto searching instead of a fixed setting, which proves the importance of DAS. All these experimental results on WSI survival prediction tasks have demonstrated the effectiveness and robustness of our proposed survival prediction scheme.

References

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

[2] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[3] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

[4] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[5] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[6] Ruoyu Li, Jiawen Yao, Xinliang Zhu, Yeqing Li, and Junzhou Huang. Graph cnn for survival analysis on whole slide pathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–182. Springer, 2018.

[7] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal 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*, pages 339–349. Springer, 2021.

[8] Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. Graphnas: Graph neural architecture

search with reinforcement learning. *arXiv preprint arXiv:1904.09981*, 2019.

[9] Kaixiong Zhou, Qingquan Song, Xiao Huang, and Xia Hu. Auto-gnn: Neural architecture search of graph neural networks. *arXiv preprint arXiv:1909.03184*, 2019.

[10] Minji Yoon, Théophile Gervet, Bryan Hooi, and Christos Faloutsos. Autonomous graph mining algorithm search with best speed/accuracy trade-off. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 751–760. IEEE, 2020.

[11] Lanning Wei, Huan Zhao, Quanming Yao, and Zhiqiang He. Pooling architecture search for graph classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2091–2100, 2021.

[12] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[14] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *stat*, 1050:21, 2016.

[15] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International conference on neural information processing*, pages 362–373. Springer, 2018.

[16] Yanxi Li, Zean Wen, Yunhe Wang, and Chang Xu. One-shot graph neural architecture search with dynamic search space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8510–8517, 2021.

[17] ZHAO Huan, YAO Quanming, and TU Weiwei. Search to aggregate neighborhood for graph neural network. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 552–563. IEEE, 2021.

[18] Yaoman Li and Irwin King. Autograph: Automated graph neural network. In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part II*, pages 189–201, 2020.

[19] Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, and Xia Hu. Policy-gnn: Aggregation optimization for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 461–471, 2020.

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

[21] Huidong Liu and Tahsin Kurc. Deep learning for survival analysis in breast cancer with whole slide image data. *Bioinformatics*, 2022.

[22] Donglin Di, Shengrui Li, Jun Zhang, and Yue Gao. Ranking-based survival prediction on histopathological whole-slide images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 428–438. Springer, 2020.

[23] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7234–7242, 2017.

[24] Manuel Muñoz-Aguirre, Vasilis F Ntasis, Santiago Rojas, and Roderic Guigó. Pyhist: a histological image segmentation tool. *PLoS computational biology*, 16(10):e1008349, 2020.