Sparse and Hierarchical Transformer for Survival Analysis on Whole Slide Images

Rui Yan, Zhilong Lv, Zhidong Yang, Senlin Lin, *Graduate Student Member, IEEE*, Chunhou Zheng, and Fa Zhang, *Member, IEEE*

***Abstract*—The Transformer-based methods provide a good opportunity for modeling the global context of gigapixel whole slide image (WSI), however, there are still two main problems in applying Transformer to WSI-based survival analysis task. First, the training data for survival analysis is limited, which makes the model prone to overfitting. This problem is even worse for Transformer-based models which require large-scale data to train. Second, WSI is of extremely high resolution (up to 150,000 × 150,000 pixels) and is typically organized as a multi-resolution pyramid. Vanilla Transformer cannot model the hierarchical structure of WSI (such as patch cluster-level relationships), which makes it incapable of learning hierarchical WSI representation. To address these problems, in this article, we propose a novel Sparse and Hierarchical Transformer (SH-Transformer) for survival analysis. Specifically, we introduce sparse self-attention to alleviate the overfitting problem, and propose a hierarchical Transformer structure to learn the hierarchical WSI representation. Experimental results based on three WSI datasets show that the proposed framework outperforms the state-of-the-art methods.**

***Index Terms*—Hierarchical representation, pathological image analysis, sparse transformer, survival analysis.**

I. Introduction

SURVIVAL analysis is one of the important contents in evaluating cancer prognosis [1], [2]. With the powerful feature extraction capability of deep learning, richer features can be directly extracted from pathological images, resulting in better survival prediction [3], [4], [5]. However, due to the inability to model the global context of whole slide image (WSI), prediction results from the existing WSI-based survival analysis methods remain unsatisfactory.

WSI is of extremely high resolution (up to $150,000 \times 150,000$ pixels), and is typically organized as a multi-resolution pyramid [6]. Therefore, deep learning methods are not able to directly process WSI as a whole. To avoid this problem, we can first select a few representative patches (hundreds × hundreds of pixels) from each WSI, and then only fuse these selected patches to obtain WSI-level feature representation for survival analysis. Therefore, a good patch selection method, especially one that can ensure the representativeness of selected patches, is a prerequisite for WSI-based survival analysis.

Currently, four categories of methods have been proposed to fuse the few representative patches selected from each WSI: Convolutional Neural Network (CNN)-based methods, Deep Multiple Instance Learning (DMIL)-based methods, Graph Neural Network (GNN)-based methods, and Transformer-based methods. A direct method is to use patch-wise CNN to extract the feature representation vectors of selected patches independently, and then concatenate or weight these feature vectors to obtain the WSI representation for survival analysis. We refer to this type of method as CNN-based method. For example, Zhu et al. [7] proposed a survival analysis method WSISA. The WSISA first randomly selects hundreds of candidate patches from each WSI and further clusters the selected patches. Then WSISA selects clusters based on patch-wise prediction performance using CNN and aggregates the selected clusters for final prediction.

CNN-based methods lack the ability to model the contextual relationship between the selected patches. Consequently, three categories of methods emerged to solve this problem: DMIL-based methods, GNN-based methods, and Transformer-based methods. The DMIL-based methods described here mainly refer to methods that use attention mechanisms for pooling [8], [9]. A representative work of DMIL-based methods is the method proposed by Yao et al. [10]. This method firstly clusters all the patches cut out from one WSI to obtain different phenotype clusters. Then, several patches are simultaneously selected from different phenotype clusters and input to the proposed network for end-to-end training. Recurrent Neural Network (RNN)-based methods [11] are also capable of modeling short-distance relationships between patches, which have been widely applied to other WSI analysis tasks outside the field of survival analysis. However, DMIL-based and RNN-based methods lack the ability to model the topology and long-distance contexts between patches.

GNN-based methods are good at modeling the topology between patches [12], such as DeepGraphSurv proposed by Li et al. [13]. This method first randomly selects a few patches

Manuscript received 29 July 2022; revised 22 July 2023; accepted 17 August 2023. Date of publication 22 August 2023; date of current version 5 January 2024. This work was supported in part by the NSFC under Grants 61932018, 32241027, 62072441, and 62072280, and in part by the National Key Research and Development Program of China under Grant 2021YFF0704300. *(Corresponding author: Fa Zhang.)*

Rui Yan is with the School of Biomedical Engineering, University of Science and Technology of China, Hefei 230026, China, also with the Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China, also with the the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yanrui@ustc.edu.cn).

Zhilong Lv, Zhidong Yang, and Senlin Lin are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lvzhilong17g@ict.ac.cn; yangzhidong19s@ict.ac.cn; linsenlin20g@ict.ac.cn).

Chunhou Zheng is with the School of Artificial Intelligence, Anhui University, Hefei 230601, China (e-mail: zhengch99@ahu.edu.cn).

Fa Zhang is with the School of Medical Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: zhangfa@ict.ac.cn).

Digital Object Identifier 10.1109/JBHI.2023.3307584

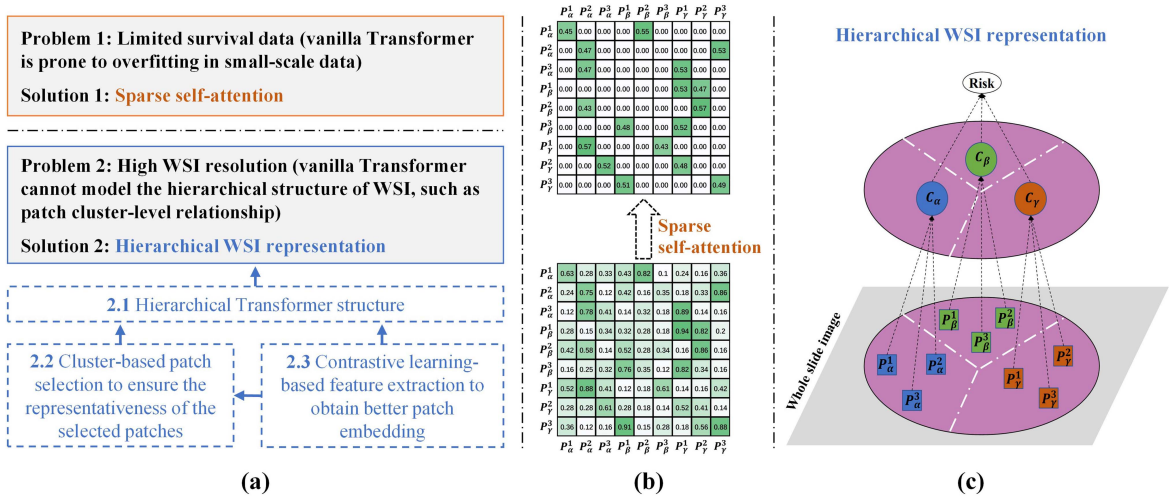

Fig. 1. Main contributions and key ideas of this article. (a) The main contributions. (b) It is not necessary for a patch in the input sequence to establish context relationship with all other patches, but only to establish relationship with some of the key patches. Here, we take the number of key patches as 2, and use the attention score matrix to represent the relationship between patches. The colored squares mean corresponding attention scores are calculated and the blank squares mean the attention scores are discarded. See Figs. 2(c) and 5 for a more detailed explanation. (c) The figure shows that WSI is clustered into 3 clusters (divided by white dotted lines), and 3 patches are randomly selected from each cluster. Embedding of selected patches are denoted by P. Cluster-level feature representations are denoted by C. See Fig. 4 for a more detailed explanation.

from each WSI to construct a graph, and then uses graph convolution network to perform graph classification task on the constructed graph. Wang et al. [14] proposed a hierarchical GNN to jointly explore patch- and cell-based topology. However, the graph structure is often manually constructed with similarity or distance metrics, which cannot necessarily reflect the real topology between patches. Once the constructed graph structure is inaccurate, it will bring bias to the subsequent analysis.

Transformer is effective at modeling long-distance dependencies between input sequences [15], [16], [17]. Therefore, Transformer-based methods provide a good opportunity for modeling the global context of WSI [18], [19], [20], [21], [22]. For example, Huang et al. [23] proposed a survival analysis method called SeTranSurv which extracts patch features through contrastive learning and fuses these features using the Transformer. Moreover, the learned self-attention weights can be considered as an adjacency matrix of the graph, which can also be understood as a topology obtained through learning [24]. Therefore, the Transformer-based method is the most promising WSI-based survival analysis method among the above four categories.

However, there are still two main problems that hinder the Transformer from being applied to WSI-based survival analysis task. First, the training data for survival analysis is limited, which makes the model prone to overfitting. This problem is even worse for Transformer-based models which require large-scale data to train. Second, the resolution of WSI is extremely high. Vanilla Transformer cannot model the hierarchical structure of WSI (such as patch cluster-level relationships, Fig. 1(c)), which makes it incapable of learning hierarchical WSI representation.

In this article, we propose a novel **S**parse and **H**ierarchical **Transformer** (**SH-Transformer**) that can model global contextual information of WSI for survival analysis, and the main contributions are summarized as follows (see Fig. 1(a)):

1) We introduce sparse self-attention to alleviate the Transformer model overfitting.
2) We propose a new framework to learn hierarchical WSI representation, consisting of three parts: **(i)** We propose a hierarchical Transformer structure to better learn patch cluster-level relationships, thereby learning hierarchical WSI representation; **(ii)** Different from randomly selecting patches from WSI as the input of the Transformer, we use the clustering method to ensure the representativeness of the selected patches; **(iii)** We adopt the contrastive learning method to extract patch features, so as to obtain better patch embedding for clustering and Transformer model.
3) Experimental results on three WSI datasets demonstrate that the proposed method outperforms the state-of-the-art methods. The C-index values are: 0.743 for bladder urothelial carcinoma, 0.739 for breast invasive carcinoma, and 0.726 for lung squamous cell carcinoma.

II. Related Work

A. Deep Learning-Based WSI Analysis

Although this article focuses on WSI-based survival analysis task, the WSI analysis methods involved in different tasks are generic. To introduce related work more comprehensively, we summarize deep learning (DP)-based WSI analysis methods involved in various tasks such as cancer classification, cancer grading, gene mutation prediction, molecular subtype prediction, and survival prediction, etc.

Image analysis methods based on DP have made significant progress in natural and radiological images [25], [26], [27], but they lag behind in WSI. The main reason for this is the resolution of WSI is too high, and general DP methods cannot be directly used on WSI. Until recently, some DP-based WSI analysis methods have gradually emerged. For the brevity of description, in this article, patch is used to represent image of equal size (usually 512×512 pixels) cut from WSI in a non-overlapping manner.

The research on DP-based WSI analysis methods can be summarized into three progressive stages. At the first stage, the community reduced the WSI analysis problem to the patch

analysis problem, in which the patch is obtained by manual selection. At this point, general DP methods can be directly applied to the patch. The DeepConvSurv model proposed by Zhu et al. [28] is a pioneering work for DP-based survival prediction using pathological images. However, this method does not meet the expectations of automated WSI analysis. In practical application, if the pathologist has located a single patch in the WSI, the preliminary diagnosis of the case is almost made. Therefore, the automatic analysis of the single patch can only play a limited role in auxiliary diagnosis.

At the second stage, the WSI analysis method based on majority voting (MV) is the most widely used framework [3], [29], [30], [31], [32], [33]. The MV described here is a general term, and MV can also be replaced by methods such as averaging, median, top ranking, etc., or an ensemble of these methods. Due to the large resolution of WSI, general DP methods cannot directly use the entire WSI (or region of interest) as input for end-to-end training. Therefore, the straightforward approach is to first cut the WSI into patches, and then use DP methods to process the patches, resulting in patch-level results. Finally, MV is used on patch-level results to get WSI-level results.

The MV-based WSI analysis method only processes each patch independently, which leads to the loss of the relationship between patches, so that the third stage method appears. Each WSI can spawn tens of thousands of patches, and it is technically difficult to directly integrate all these patches end-to-end. The best practice [3], [5], [7], [10], [11], [13] is to select a few representative patches (assuming there are hundreds of Selected-Patches, abbreviated as SP) first, and then fuse these SP to get the final prediction result. The process of selecting SP from a WSI is similar to the process of text summarization [34] or video summarization [35], which can be named as WSI summarization.

Most of the work [4], [13] initially degenerates into randomly generating SP directly from the WSI (or region of interest), however, randomly generated SP are not representative enough. To deal with this problem, patch selection is performed through patch-level prediction results [3], [7], patch-level prediction probability heatmaps [5], [11], clustering-based methods [7], [10], etc.

After generating SP, the next step is to fuse these SP for WSI prediction. Two fusion methods exist: one-step fusion method and two-step fusion method. The one-step fusion method directly uses SP as input, and then trains it in end-to-end manner. These end-to-end methods include CNN-LSTM [11], GCN [36], and DMIL [10], etc. The two-step fusion method first extracts the feature representation vectors of SP independently, and then fuses these vectors to obtain WSI-level results. These fusion methods include CNN [37], LSTM [38], GNN [39], DMIL [40], and Transformer [18], [19], [20], [21], [23], etc. In general, the two-step fusion method is more flexible than the one-step fusion method since there is no large memory requirement for end-to-end training. However, none of these SP fusion methods can learn the hierarchical feature representation of gigapixel WSI.

B. Application of Transformer on WSI

Transformer was originally proposed by Vaswani et al. [41] for machine translation task. Subsequently, Transformer-based pre-trained models [42] achieve significant breakthroughs on various natural language processing (NLP) tasks [43], [44]. For example, Devlin et al. [43] proposed a language representation model called BERT. It achieved excellent performance on 11 different types of downstream NLP tasks.

As a result of Transformer architecture's success in NLP, researchers have recently applied it to computer vision tasks [15]. Vision Transformer (ViT) proposed by Dosovitskiy et al. [45] directly applies a pure Transformer to the sequences of image patches, it has achieved state-of-the-art performance on multiple image classification benchmarks. Based on ViT, Swin Transformers [46] introduced priors such as hierarchy, locality and translation invariance into the Transformer network structure design to achieve better performance in multiple visual tasks, and its computational complexity is linearly related to image size. Hassani et al. [47] proposed the Neighborhood Attention Transformer (NAT), a new efficient, accurate, and scalable hierarchical Transformer composed of Neighborhood Attention (NA). NA can localize the receptive field for each token to its neighborhood, and as the receptive field increases, the effect of NA is close to self-attention.

Taking advantage of Transformer's ability to capture long-range contexts, Transformer-based methods have gradually gained popularity in the field of WSI analysis [18], [19], [20], [48], [49]. To capture local structure and global context information of histopathological images simultaneously, the TransPath method proposed by Wang [20] integrates CNN and Transformer. Chen et al. [48] proposed the multi-scale GasHis-Transformer to classify gastric histopathological image into abnormal and normal. This approach considers both the advantages of the CNN model in describing local information and the Transformer model in describing global information. The DT-MIL method proposed by Li et al. [18] introduces the deformable Transformer encoder and decoder architecture to obtain bag embedding for generating WSI-level bag representation. Shao et al. [19] proposed a Transformer-based MIL (TransMIL) to fully explore both morphological and spatial information between different instances. The instances here refer to patches sampled from WSI. TransMIL can achieve better performance and faster convergence compared with state-of-the-art methods.

Self-attention is an essential component of Transformer, but it also poses one major challenge in practice: quadratic complexity of the input sequence length. Therefore, the self-attention becomes a bottleneck when dealing with long sequences or small dataset. Especially when Transformer is applied to gigapixel WSI analysis tasks, the insufficient number of selected patches will result in poor representation of the information contained in WSI. Inspired by the observation that the learned attention matrix of the Transformers is often very sparse across most data points [50], it is possible to reduce computation complexity by incorporating structural bias to limit the number of query-key pairs. For example, global attention pattern [51] adds some global nodes as the hub for information propagation between nodes, band attention pattern [52] restricts each query to attend to its neighbor nodes, and random attention pattern [53] randomly samples a few edges for each query to increase the ability of non-local interactions. However, they either weaken long-term dependence or impair the self-attention mechanism's focus on the key parts. In this study, we introduce a sparse self-attention method to solve this problem in applying Transformer to WSI.

III. Methods

An overview of our proposed WSI-based survival analysis framework is shown in Fig. 2. We will introduce this framework from 5 parts: the patch embedding based on unsupervised

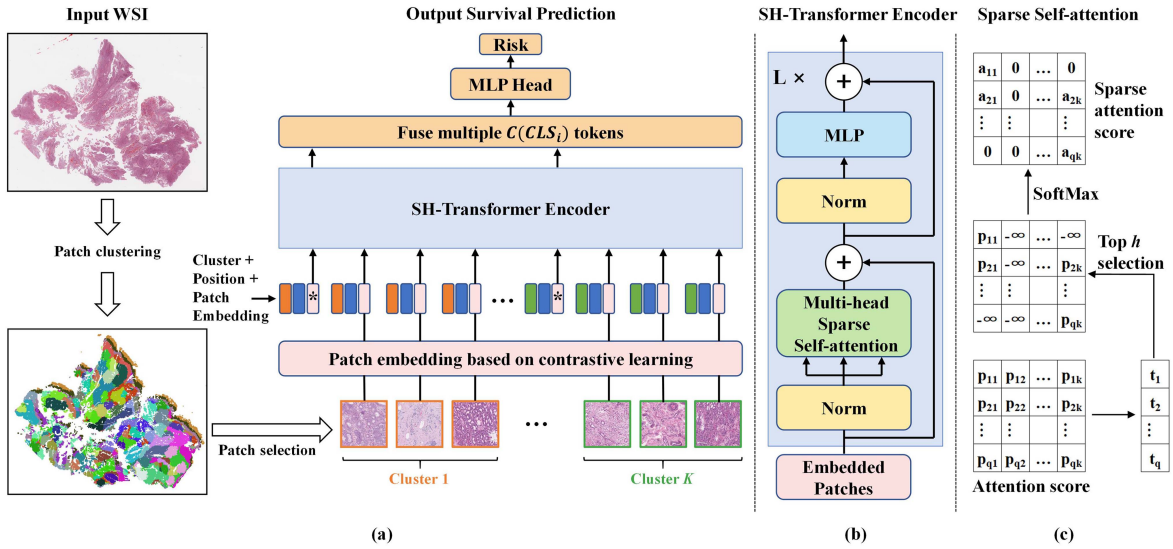

Fig. 2. Overview of our proposed framework. (a) The framework consists of three components: patch embedding based on unsupervised contrastive learning, patch selection based on clustering, and the SH-Transformer. The blue-filled rectangles represent the positional embeddings, and the yellow and green-filled rectangles represent the clustering embeddings, respectively. '$*$' represent learnable embedding of [CLS] token. (b) Network details of SH-Transformer encoder. (c) The illustration of the sparse self-attention.

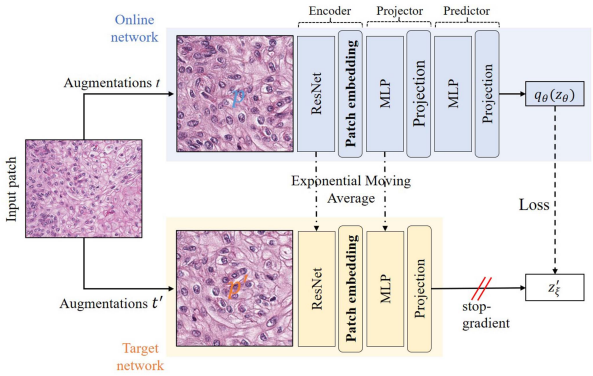

Fig. 3. BYOL's architecture used in this study.

contrastive learning, the patch selection based on clustering, the architecture of SH-Transformer, the sparse self-attention, and the loss function used in network training.

A. Patch Embedding Based on Contrastive Learning

Since it is difficult to obtain patch-level survival annotations, patch embedding cannot be performed using supervised learning methods. Recently, the self-supervised contrastive learning method has made great progress in feature extraction, so we choose to use it for patch embedding.

Bootstrap Your Own Latent (BYOL) [54], [55] is a type of contrastive learning method. The comparison of the feature representation learning ability of different contrastive learning methods are summarized in Table III. BYOL contains two networks: the online and target networks. The online network is defined by a set of parameters θ, as shown in Fig. 3. The target network has the same network structure as the online network, but has a different set of parameters ξ. The target network provides the targets to train the online network, and its parameters ξ are an exponential moving average of θ. Specifically, given a target decay rate $\tau \in [0,1]$, after each training step, the parameters ξ are updated by:

$$\xi \leftarrow \tau\xi + (1-\tau)\,\theta, \tag{1}$$

the loss function is defined as mean squared error between the normalized predictions and target projections:

$$L_{\theta,\xi} \triangleq \left\| \frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|_2} - \frac{z_\xi{}'}{\|z_\xi{}'\|_2} \right\|_2^2. \tag{2}$$

Similar to loss $L_{\theta,\xi}$ in (2), we separately feeding p to the target network and p' to the online network to compute $\bar{L}_{\theta,\xi}$. The overall optimization goal is to minimize $L_{\theta,\xi}^{BYOL} = L_{\theta,\xi} + \bar{L}_{\theta,\xi}$ with respect to θ only, but not ξ. After the network training converges, the "Patch embedding" part of online network is used as the pathological image representation.

B. Patch Selection Based on Clustering

The self-attention module has the quadratic complexity of the input sequence length, which limits the input sequence length of the Transformer [24]. For example, the maximum input sequence length allowed by the original BERT [43] is 512. Tens of thousands of patches can be cut out from one WSI, which far exceeds the processing capability of the Transformer. To apply the Transformer, we can select only a few patches from each WSI at one time. However, if the patch selection is random, the representativeness of the selected patches cannot be guaranteed.

To ensure the representativeness of the selected patches, we first cluster all the patches cut out from each WSI, and then select a fixed number of patches from each cluster. In this way, survival-related patches are selected with a higher probability. Specifically, based on BYOL mentioned above, each patch can extract a 2048-dimensional feature representation vector for patch clustering. Next, we cluster the patches into K clusters using the K-Means algorithm, and then randomly selected N patches from each cluster, that is, $K \times N$ patches are selected from one WSI each time.

C. SH-Transformer Architecture

The proposed SH-Transformer addresses the two main problems that hinder the use of Transformer in WSI-based survival

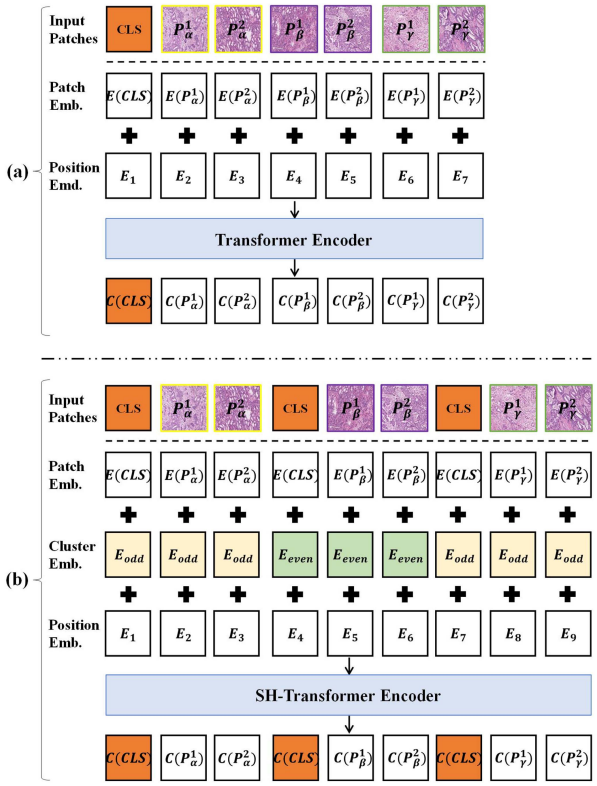

Fig. 4. Architecture of the vanilla ViT (a) and SH-Transformer (b). The sequence on top is the input patches select from the different clusters of each WSI. The summed vectors of three kinds of embeddings are used as input embeddings to Transformer encoder.

analysis tasks: model overfitting and the inability to learn hierarchical WSI representation. As a result, the global context information of WSI can be better modeled.

The SH-Transformer (Fig. 2(a)) is an improvement based on the Vision Transformer (ViT) [45]. The architecture comparison of the vanilla ViT and SH-Transformer is shown in Fig. 4. The SH-Transformer input is a sequence of $K \times N$ patches: $[P_1^1, P_1^2, \ldots, P_1^N; P_2^1, P_2^2, \ldots, P_2^N; \ldots; P_K^1, P_K^2, \ldots, P_K^N]$, where K is the number of clusters per WSI, and N is the number of patches selected from each cluster. The patch embedding operation is denoted as $E(.)$. Similar with patch clustering, the patch embedding here is also obtained using BYOL method.

The vanilla ViT prepends a learnable embedding ($[CLS]$ token) as global nodes to the sequence of embedded patches. The state of $C(CLS)$ at the output of the Transformer encoder is used as the WSI representation, as shown in Fig. 4(a). The WSI is divided into different clusters, and the vanilla ViT can only learns the short- and long-distance relationship between patches, but cannot model the relationship between clusters. To better learn intra- and inter-cluster relationships, thereby learning hierarchical WSI representations, we introduce additional token embeddings, as shown in Fig. 4(b). Specifically, to represent individual cluster, we insert multiple external $[CLS]$ tokens at the start of each cluster, and each $[CLS]$ symbol collects features for the patches contained in this cluster. We also use interval segment embeddings to distinguish multiple clusters within a WSI. For $cluster_k$, we assign learnable segment embedding E_{odd} or E_{even} depending on whether k is odd or even. For example, when $k = 5$, we assign embeddings $[E_{odd}, E_{even}, E_{odd}, E_{even}, E_{odd}]$. In this way, WSI representations are learned hierarchically where lower Transformer layers represent adjacent clusters, while higher layers represent multi-cluster. This method is mainly inspired by the BERT-based text summarization method proposed by Liu et al. [56]. Similar to text summarization, we can consider the process of obtaining WSI representation as WSI summarization, in which **WSI** is regarded as "paragraph" **cluster** is regarded as "sentence" and **patch** is regarded as "word".

Since the Transformer model is ignorant of positional information, additional positional representation is needed to model the sequence order information. We use standard learnable 1D position embeddings, because we do not get performance boost when using more advanced 2D position embeddings. Therefore, the summed vectors z_0 of three kinds of embeddings (i.e., Patch Embedding + Cluster Embedding + Position Embedding) are used as input embeddings to the SH-Transformer encoder. The output of the SH-Transformer encoder is the generated vector containing rich contextual information for each input patch embedding. We fuse multiple $C(CLS_i)$ as the hierarchical WSI representation z_L. It goes through an MLP module to obtain a predicted risk value.

The SH-Transformer encoder [41] consists of multi-headed sparse self-attention blocks (MSSA) and multi-layer perceptron (MLP) blocks, as shown in Fig. 2(b). We will introduction sparse self-attention in the next subsection in detail. The output of the l-th SH-Transformer encoder is defined as z_l. LayerNorm (LN) operation is applied before every block, and residual connection is added after every block. The MLP head consists of fully connected layer with one hidden layer. The SH-Transformer can be formulated into:

$$z_0 = [E(CLS_k); E(P_k^n)] + E_{cluster} + E_{pos}, \quad k = 1 \ldots K, \;\; n = 1 \ldots N, \tag{3}$$

$$z'_l = MSSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \ldots L, \tag{4}$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1 \ldots L, \tag{5}$$

$$risk = MLP(z_L). \tag{6}$$

D. Sparse Self-Attention

The vanilla self-attention (SA) in Transformer encoder is defined as follows:

$$P = \frac{Q^\top}{\sqrt{d}}, \tag{7}$$

$$Att(Q, K, V) = \text{softmax}(P)V, \tag{8}$$

where Q, K, and V denotes the queries, keys, and values matrices respectively, d is the dimension of queries and keys, P is the attention scores. Multi-headed self-attention (MSA) is an extension of SA that performs multiple SA operations in parallel, and projects their concatenated outputs.

Transformer is prone to overfitting in small-scale data due to its lack of structural bias. Additionally, datasets for survival analysis are difficult to collect, which further leads to model overfitting. To alleviate this problem, we introduce sparse self-attention.

For our WSI analysis problem, it is not necessary for a patch in the input sequence to establish a contextual relationship with all other patches, but only to establish a relationship with some of the key patches. This can be understood as the network topology

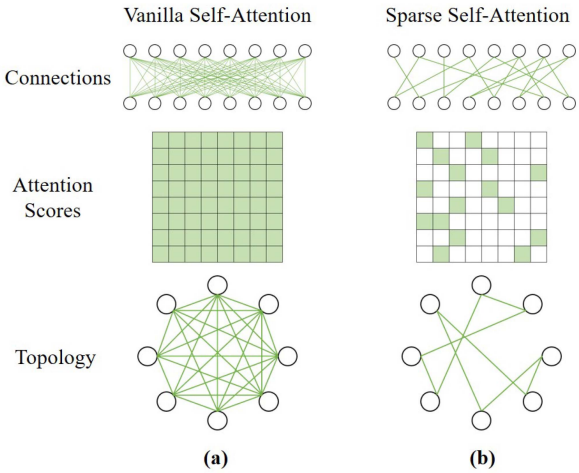

Fig. 5. Comparison of different topologies. The circle nodes indicate the hidden states of input tokens. The colored squares mean corresponding attention scores are calculated and the blank squares mean the attention scores are discarded. (a) The vanilla self-attention can be regarded as a complete bipartite graph where each query receives information from all memory nodes and updates its representation. (b) The sparse self-attention can be considered as a sparse graph where some of the connections between nodes are removed. The learned explicit sparse self-attention scores can be considered as an adjacency matrix of the graph, which can also be understood as a learned topology [24].

we need is a partially connected graph (see Fig. 5(b)), rather than a fully connected graph (see Fig. 5(a)). We believe this is one of the main reasons for the model overfitting in Transformer. After sparse attention, the obtained partially connected graph is equivalent to adding additional regularization constraints to the model, thus alleviating its overfitting. This is similar to the role of "Dropout" [57], [58] in mitigating overfitting in general neural networks.

In this article, we use the explicit sparse Transformer proposed by Zhao et al. [59] to deal with this problem. The illustration of the sparse attention module is in Fig. 2(c). Through top-h selection, not only the most contributive components for attention are reserved and the other irrelevant information or noise are removed, but also the model is simplified since h is usually a small number such as 16.

To select the top-h contributive elements of P, we conduct sparse attention masking operation $M()$ which is illustrated as follows:

$$M(P,h)_{ij} = \begin{cases} P_{ij}, & \text{if } P_{ij} \geq t_i, \\ -\infty, & \text{if } P_{ij} < t_i. \end{cases} \tag{9}$$

where h is a hyperparameter, and t_i denotes the h-th largest value of row i. After the step of top-h selection, the sparse self-attention is defined as follows:

$$SparseAtt(Q,K,V,h) = \text{softmax}(M(P,h))V. \tag{10}$$

E. Loss Function

Similar to the classic survival analysis methods Cox proportional hazard model [60], the network optimizes a negative partial log likelihood as the loss function, which can be expressed as:

$$L(R) = \sum_{i \in \{i:S_i=1\}} \left(-R_i + log \sum_{j \in \{j:T_j \leq T_i\}} (exp(R_j)) \right), \tag{11}$$

TABLE I
DATASET STATISTICS

Subtype	# Cases	# Deaths	# WSIs	# Groups	Magnif.
BLCA	386	175	452	9,040	40X
LUSC	475	205	508	10,160	40X
BRCA	982	146	1,016	15,240	40X

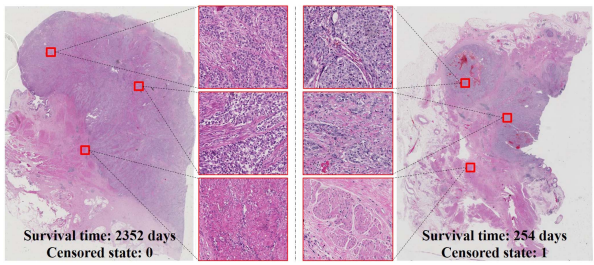

Fig. 6. Examples of WSI with different survival time and censored state.

where R represents the final output of the framework, which is the patient's predicted risk. S_i, T_i are the censoring status and the survival time of i-th patient, respectively. $S_i = 1$ represents death observed.

F. Complexity Analysis

The main difference between our proposed SH-Transformer and vanilla ViT is in two places: (i) sparse self-attention, and (ii) token embedding. (i) Assuming $Q, K, V \in \mathbb{R}^{T \times D}$, the complexity of vanilla self-attention $(softmax(QK^T)V)$ is $\mathcal{O}(T^2 \cdot D)$, T represents sequence length, and D represents the representation dimension [24]. Similarly, the complexity of sparse self-attention $(softmax(M(P,h))V)$ is also $\mathcal{O}(T^2 \cdot D)$. The sparse self-attention operation is equivalent to adding top-h selection on the basis of the vanilla self-attention calculation, and these calculations are negligible. (ii) In addition, in the process of token embedding, only element-by-element vector addition is performed ($z_0 = [E(CLS_k); E(P_k^n)] + E_{cluster} + E_{pos}$), and the calculation are also negligible. In summary, the time complexity of SH-Transformer and vanilla ViT is basically the same. When the number of Transformer encoder blocks $L = 6$ and the input sequence is 500 patches (the patch embedding dimension is 1280), compared with the vanilla ViT (118 M), the total parameters of our proposed SH-Transformer (120 M) only increase by 1.7%.

IV. EXPERIMENTS

A. Dataset Description

We conducted experiments on three WSI datasets: Bladder urothelial carcinoma (BLCA), Breast invasive carcinoma (BRCA), and Lung squamous cell carcinoma (LUSC), all of which are from The Cancer Genome Atlas (TCGA) [61]. The dataset can be obtained from https://gdc.cancer.gov. The numbers of WSIs, patients, and groups in each dataset are shown in Table I. In this study, one WSI is clustered into 50 clusters, and 11 patches (512×512 pixels, level $= 0$) are randomly selected from a cluster, so that 550 patches are selected from one WSI to form a group. Repeating this random selection S times, a total of S groups will be obtained from one WSI. The examples of WSI with different survival time and censored state are shown in Fig. 6.

TABLE II
PERFORMANCE COMPARISON USING C-INDEX VALUES (MEAN AND STANDARD DEVIATION) ON THREE DATASETS

Methods	BLCA	LUSC	BRCA
WSISA [7]	0.617±0.006	0.612±0.004	0.637±0.003
WSISA + BYOL	0.636±0.005	0.638±0.003	0.642±0.005
DeepAttnMISL [10]	0.682±0.011	0.670±0.011	0.675±0.010
DMIL + BYOL + K-Means	0.683±0.007	0.672±0.008	0.679±0.007
DeepGraphSurv [13]	0.658±0.018	0.647±0.020	0.674±0.022
DeepGraphSurv + BYOL	0.662±0.016	0.660±0.018	0.682±0.016
DeepGraphSurv + BYOL + K-Means	0.674±0.014	0.668±0.012	0.690±0.014
Huang et al. [23]	0.700±0.008	0.701±0.010	0.705±0.010
Transformer + BYOL	0.705±0.010	0.704±0.012	0.709±0.013
Transformer + BYOL + K-Means	0.716±0.008	0.712±0.006	0.711±0.006
Transformer + BYOL + K-Means + Sparse	0.732±0.005	0.721±0.008	0.723±0.005
SH-Transformer (Ours)	**0.743±0.006**	**0.726±0.007**	**0.739±0.006**

The bold values indicate the highest values.

B. Baseline Methods and Evaluation Metric

1) Baseline Methods: In our experiments, the following state-of-the-art WSI-based survival analysis methods without using region-of-interest annotations or patch-wise annotations are compared. Survival analysis methods based on WSI can be summarized into four categories: CNN-based methods, GNN-based methods, DMIL-based methods, and Transformer-based methods. We conduct experimental comparisons with representative methods in each category. The methods marked with "+BYOL" mean that patch features are extracted using the self-supervised contrastive learning method BYOL. Similarly, the methods marked with "+K" mean patch selection using clustering methods K-Means.

- CNN-based methods: WSISA [7]; WSISA + BYOL.
- GNN-based methods: DeepGraphSurv [13]; DeepGraphSurv + BYOL; DeepGraphSurv + BYOL + K.
- DMIL-based methods: DeepAttnMISL [10]; DMIL [8] + BYOL + K.
- Transformer-based methods: Huang et al. [23].

2) Evaluation Metric: To evaluate the performance of our framework, we mainly use the Concordance index (C-index) [62], which measures the concordance of ranking of predicted risk with the ground truth survival times. The C-index is calculated as follows:

$$C_{index} = \frac{1}{n} \sum_{i \in \{1 \ldots n | \delta_i = 1\}} \sum_{t_j > t_i} I\left[f_i > f_j\right], \tag{12}$$

where n is the number of comparable pairs and $I[\cdot]$ is the indicator function. t is the actual survival time observation. f denotes the corresponding predicted risk. The C-index value ranges between 0 and 1. The C-index value of 0.5 indicates ineffective prediction. The higher the C-index, the better the survival prognosis. In addition to the C-index metric, we also use two other metrics: time-dependent Receiver Operating Characteristic (ROC) curve (with the AUC values) and univariate KM-estimation (with Log-rank p-values).

C. Implementation Details

The datasets are randomly split into training, validation, and testing sets (70% : 15% : 15%). The BYOL uses ResNet50 as CNN backbone, and uses Adam optimizer with an learning rate of 0.0001. For K-means, we take $K = 50$. Since our K value is large, we increase the number of times the algorithm is run with different initialization centroids to 20. For Transformer, we take epoch $= 1000$, top-$h = 16$, no pretrained model, an learning rate of 0.01. For training SH-Transformer, we perform data augmentation. Specifically, one WSI is clustered into 50 clusters, and 11 patches (512×512 pixels) are randomly selected from a cluster, so that 550 patches are selected from one WSI. Repeating this random selection S times, a total of S groups will be obtained from one WSI. In this way, the total sample size has increased S times. In this work, we set $S_{BLCA} = 20$, $S_{LUSC} = 20$, and $S_{BRCA} = 15$. All experiments in this article are implemented with scikit-learn and PyTorch framework, and finished on four NVIDIA GeForce GTX 3090 GPUs.

D. Overall Predictive Performance

The performance of our proposed framework is shown in Table II. We conduct each experiment by using holdout cross-validation. In particular, the datasets (group-level) partitioned into training set and testing set. The random partitioning process is repeated 10 times and the mean and standard deviation are reported. We compare the survival analysis performance with state-of-the-art methods. Our framework achieves the best C-index value on all three datasets. Among the four categories of WSI-based survival analysis methods, CNN-based methods performed suboptimally. This is primarily due to the fact that it does not make good use of the relationship between patches. In contrast, the GNN-based method and the DMIL-based method have similar prediction performance and both yield better results, and Transformer-based methods show obvious advantages and achieve the best prediction performance.

To evaluate the effectiveness and generality of each component in our proposed framework, we conduct a comparative experiment in all methods on whether use patch embedding based on contrastive learning (BYOL) or patch selection based on clustering (K-means). It can be seen from the results in Table II that after adding BYOL and K-means to the original method, the prediction performance of all methods has been improved. Specifically, when K-means is not used, we randomly select 550 patches from the entire WSI each time for SH-Transformer training and prediction.

In addition, we also verify the effectiveness of sparse self-attention and hierarchical WSI structure. It can be seen from the experimental results that the best prediction results are achieved when these two components are used simultaneously. The experimental results show that sparse self-attention can significantly reduce the problem of model overfitting, which is a prerequisite for the model to learn the hierarchical WSI representation. In other words, when the model only uses the hierarchical WSI structure but not uses sparse self-attention, the overfitting problem of the model still exists, resulting in no performance improvement for the model.

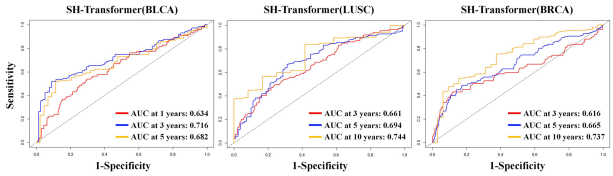

Fig. 7. Time-dependent ROC curves and AUC values of SH-Transformer on three datasets.

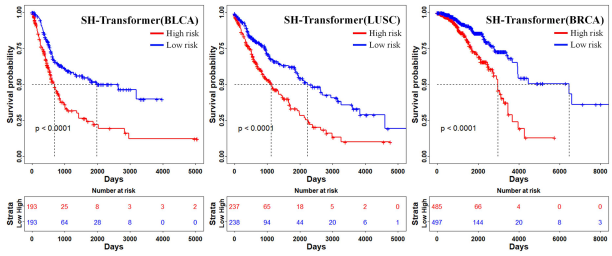

Fig. 8. Kaplan-Meier curves of SH-Transformer on three datasets.

E. The ROC and Kaplan-Meier Curves

To comprehensively demonstrate the predictive performance of our proposed method, two more metrics are adopted. First, following DeepAttnMISL [10], the time-dependent ROC curve and AUC value that evaluates the performance of time-specific survival probability prediction is also reported. The time-dependent ROC curves and AUC values of different methods for TCGA-BLCA WSI dataset are shown in Fig. 7. To further compare the survival prediction effects of different methods, we show the time-dependent ROC curves and AUC values of different methods for TCGA-BLCA WSI dataset, as shown in Fig. 9. It can be seen from the results that SH-Transformer has achieved better prediction results than the other three methods. Secondly, to evaluate whether our proposed framework can capture the overall prognosis difference and stratify patients into low- and high-risk groups, we draw Kaplan-Meier curves of SH-Transformer on all three datasets, as shown in Fig. 8. All samples are divided into high and low groups based on the median of risk score predicted by SH-Transformer. It is shown that SH-Transformer is superior both from Log-rank p-values and the significant differences in curves between the two groups. To further compare the survival prediction effects of different methods, we performed Kaplan-Meier survival curves for TCGA-BLCA WSI dataset using different methods, as shown in Fig. 10. It can be seen from the results that SH-Transformer can better distinguish high and low risk patients.

F. Patch Clustering Result

Before performing unsupervised patch clustering, we first convert patches into feature representation vectors using an unsupervised contrastive learning method. According to the results reported in literature [54], [55], we first select three contrastive learning methods (SimCLR-V2 [63], SwAV [64], BYOL [55]) with the best performance at present. Then, we validated these methods on the task of pathological image classification. The dataset we used is one we built to perform cancer and normal pathological image classification tasks. We evaluate the feature representation learning ability of different contrastive learning methods by training a fully connected (FC) network classifier on top of the frozen representation. Specifically, after the training convergence of the contrastive learning model, we use this model to extract the feature vector of the image, and then use the FC network for classification. From the results (see Table III), we find that BYOL can achieve the best unsupervised feature extraction effect on pathological images.

TABLE III
COMPARISON OF THE FEATURE REPRESENTATION LEARNING ABILITY OF DIFFERENT CONTRASTIVE LEARNING METHODS ON PATHOLOGICAL IMAGES, FC REPRESENTS FULLY CONNECTED NETWORK

Method	Accuracy (%)
SimCLR-V2 [65] + FC	78.6
SwAV [66] + FC	80.4
BYOL [57] + FC	81.3

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT CLUSTERING METHODS

Clustering method	SC	CH
Spectral Clustering	0.11	351.2
Agglomerative Clustering	0.12	913.8
Affinity Propagation	0.18	788.2
K-means	**0.21**	**1208.2**

The bold values indicate the highest values.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT NUMBER OF CLUSTERS

K-means	SC	CH
K = 30	0.13	1032.3
K = 50	**0.21**	**1208.2**
K = 70	0.18	1182.6
K = 90	0.16	992.4
K = 110	0.16	926.6

The bold values indicate the highest values.

After choosing to use BYOL for patch embedding, we experiment to choose an clustering method (see Table IV). Patches from TCGA-BLCA WSI dataset are clustered based on feature vectors extracted by BYOL. We evaluate the clustering performance of different clustering algorithms based on two metrics: Silhouette-Coefficient (SC) and Calinski-Harabasz (CH). K-means has achieved good clustering results on both metrics. It should be pointed out that there is no necessarily causal relationship between higher SC and CH values and better predictive performance. Through experimental comparison (Table II) we get the conclusion that the clustering-based patch selection strategy is effective, so we want to obtain good clustering results. High SC and CH values are just to ensure good clustering results. Also, the purpose of clustering is only for the comprehensiveness of patch selection, and as long as the patches are clustered well, different SC and CH values have no fundamental effect on the survival prediction framework.

We further investigate how many clusters to choose, that is, the choice of the hyperparameter K in K-means. Based on the metrics SC and CH, we finally choose $K = 50$. According to the human cell landscape at single-cell level proposed by Han et al. [65], the human body contains 102 types of cells, so we set the maximum number of clusters to 110. Meanwhile, if the number of clusters is too small, the representativeness of our patch selection cannot be guaranteed, so we set the minimum number of clusters to 30. Then, at every interval of 20, we report the SC and CH value of K-means (Table V).

According to the experimental results, we finally determine the combination of "BYOL + K" to cluster patches in WSI. The

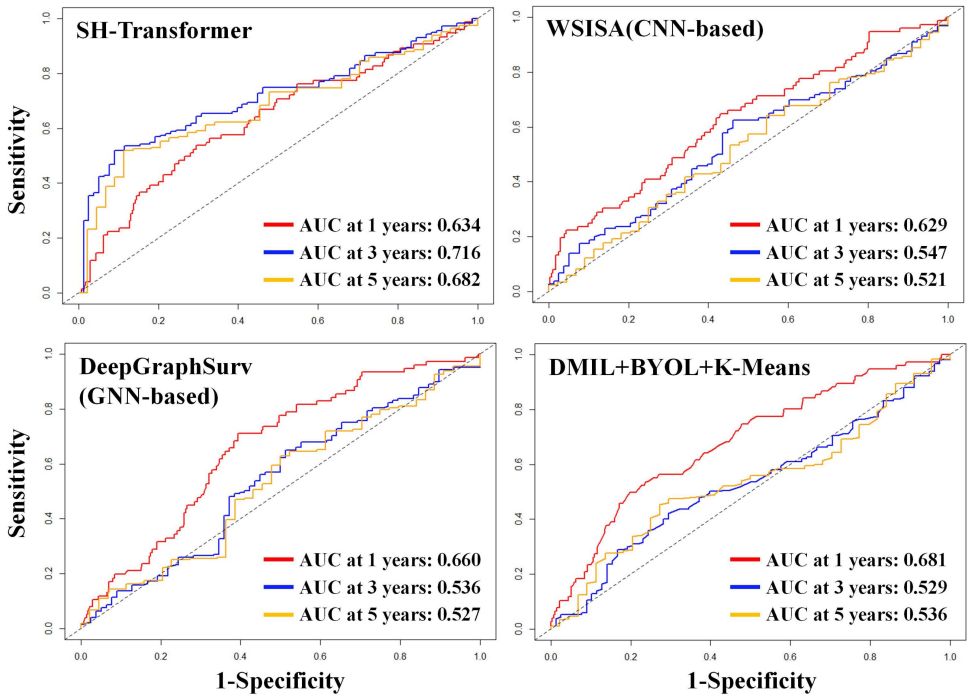

Fig. 9. Time-dependent ROC curves and AUC values of different methods on TCGA-BLCA WSI dataset.

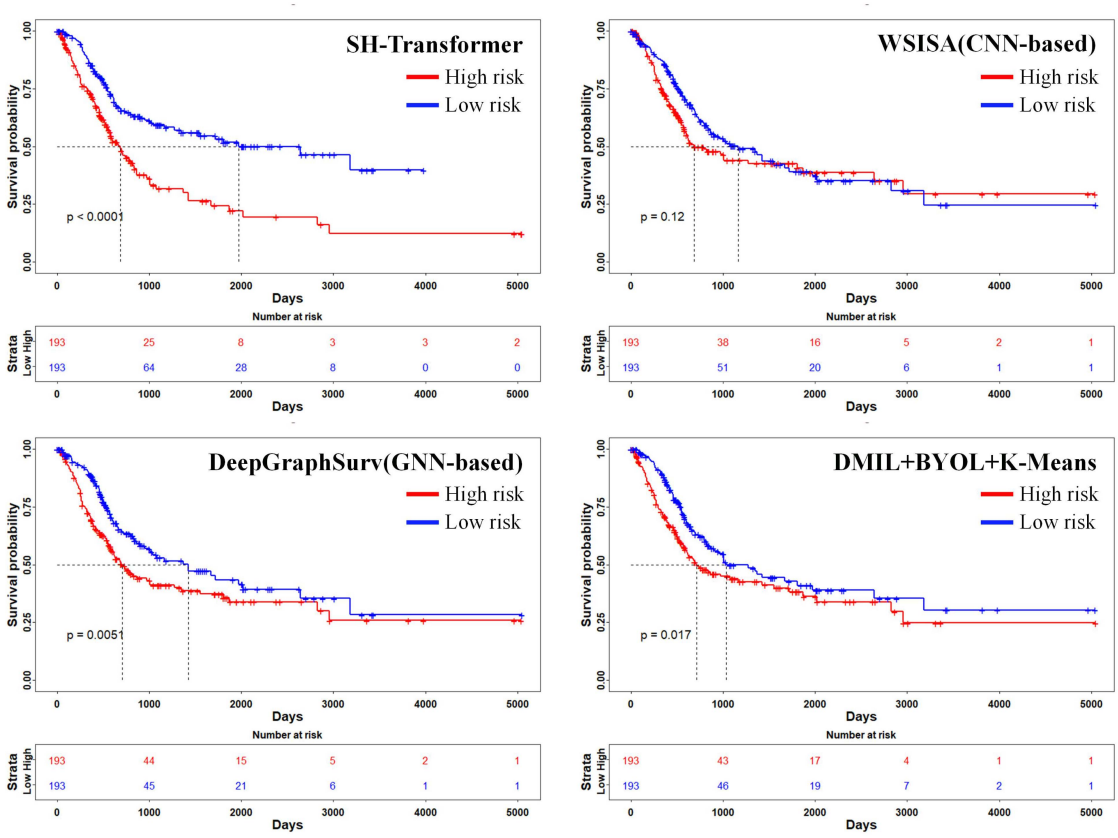

Fig. 10. Kaplan-Meier curves of different methods on TCGA-BLCA WSI dataset.

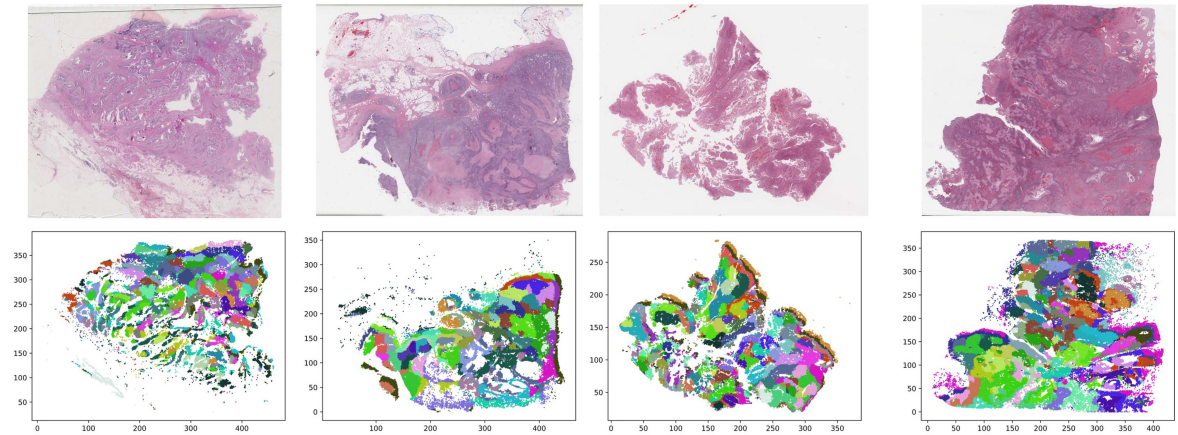

Fig. 11. Visualization of patch clustering results ($K = 50$).

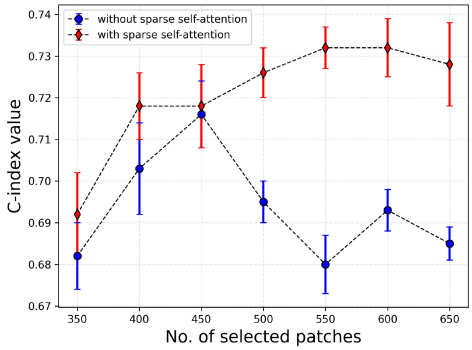

Fig. 12. Ablation study on the number of selected patches.

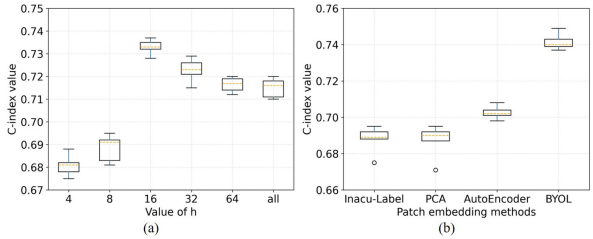

Fig. 13. Box-plot of C-index value. (a) Ablation studies on the value of h. "all" denotes all the positions is attended, i.e., no sparse self-attention is used. (b) Ablation studies on patch embedding method.

visualization of clustering results for TCGA-BLCA WSI ($K = 50$) are shown in Fig. 11. Clustering results show that spatially adjacent patches are clustered together. In addition, we can find the existence of tumor spatial heterogeneity. Clustering-based methods can select more phenotypic patches, which can better characterize tumor spatial heterogeneity.

G. Ablation Study

We conduct ablation studies to show the effect of different components of SH-Transformer. The experiments involved in this section are all performed on the TCGA-BLCA dataset. First, we study the impact of the number of patches selected from each WSI on our framework. The experimental results are shown in Fig. 12. We fix the rest of the framework (Transformer + BYOL + K), and only change the number of patches input to Transformer. It should be pointed out that in order to control variables, we do not use the hierarchical WSI representation in this comparative experiment. Overall, the model with sparse self-attention outperforms the model without it by alleviating model overfitting, especially when using long patch sequences (Fig. 12). Based on the experimental results, we have the following three observations:

1) When the number of selected patches is less than 450, the model prediction performance will improve as the number of selected patches increases. This is because more patches can reflect the information of WSI more comprehensively, so better results can be obtained. Benefiting from the mitigation of model overfitting by sparse self-attention, the prediction performance of the model with sparse self-attention is slightly higher than that of the model without sparse self-attention.
2) When the number of selected patches exceeds 450, the model prediction performance with or without sparse self-attention diverges. Unlike the model with sparse self-attention can continue to improve, the model without it starts to decrease in prediction performance. This is because long input sequences bring more serious model overfitting problem. When the number of selected patches is 550, the model (Transformer + BYOL + K + Sparse self-attention) achieved the best prediction performance with an average C-index of 0.732.
3) When the number of selected patches exceeds 550, the overall predictive performance (C-index) of the model does not continue to improve. It is not that the ability of sparse self-attention is reduced when inputting long patch sequences, but that long input patch sequences lead to more serious overfitting problems. As mentioned above, self-attention has quadratic complexity of the input sequence length. We can also understand that the overfitting problem at this time has exceeded the capability boundary of sparse self-attention.

Second, we also investigate how our framework is affected by the value of h in the Sparse Transformer, and the experimental results are shown in Fig. 13(a). Unlike the original article [59]

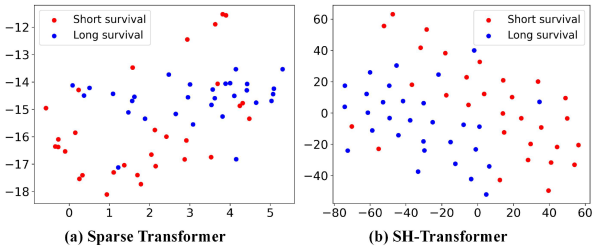

Fig. 14. t-SNE visualization of the WSI representation vectors obtained by different Transformer methods.

that proposed a Sparse Transformer with a value of 8, our framework performs at its best when the value of h is 16. When h is too small, it does not suffice to represent the input sequence because our input sequence is very long. When h is greater than 32, the performance of the model does not improve with the increase of h. Especially when h is greater than 64, the performance of the model is similar to that of not using sparse self-attention (denoted as "all" in Fig. 13(a)).

Third, to verify the effect of hierarchical WSI representation, we used the mean survival time of uncensored cases as a threshold (553 days) to classify test cases (60 patients) into long-survival time and short-survival time, and then performed t-SNE visualization [66] on the feature representation vector of WSI. Specifically, we use the single $C(CLS)$ feature vector as the WSI representation when sparse Transformer is used. Correspondingly, when using SH-Transformer, we concatenate multiple $C(CLS_i)$ as the hierarchical WSI representation, which is z_L in (6). It can be seen from Fig. 14 that the hierarchical WSI representation can better distinguish between the short and long survival time cases.

Finally, we experiment and compare different unsupervised patch embedding methods. Since patch embedding is the basis of patch clustering and Transformer, a good patch embedding method is critical to improve the overall performance of SH-Transformer. We fix other methods (Transformer + Patch-Embedding + K + Sparse + Hierarchical), and compare three representative unsupervised patch embedding methods: PCA (512D), Auto-encoder (512D), and BYOL (2048D). We have also compared it with the supervised patch embedding method (ResNet50). Specifically, after the ResNet50 training converges, we use the output of the fully connected layer (2048D) at the tail of ResNet50 as the patch embedding. For ResNet50 training, all patches cut out from one WSI inherit the same survival time label, so the label is inaccurate. It can be seen from the experimental results (Fig. 13(b)) that BYOL achieves the best prediction effect.

V. Conclusion

In this article, we proposed a new framework for WSI-based survival analysis. Our framework address two challenges in applying Transformer to WSI: (1) We introduce sparse self-attention to alleviate the overfitting problem; (2) We propose a hierarchical Transformer structure to learn the hierarchical WSI representation. In addition, to better learn the hierarchical WSI representation, we also use clustering methods to ensure the representativeness of the selected patches, and use contrastive learning methods for better patch embedding. Extensive experiments on three datasets and comparisons with the state-of-the-art methods show superior performance of our proposed framework. Our proposed framework makes Transformer-based methods feasible on WSI analysis tasks with small datasets. Moreover, our proposed framework is a general WSI representation learning framework that can be easily extended to other WSI analysis tasks.

References

[1] Z. Huang et al., "SALMON: Survival analysis learning with multi-omics neural networks on breast cancer," *Front. Genet.*, vol. 10, 2019, Art. no. 166.
[2] R. J. Chen et al., "Pathomic fusion: An integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis," *IEEE Trans. Med. Imag.*, vol. 41, no. 4, pp. 757–770, Apr. 2022.
[3] R. Yamashita et al., "Deep learning model for the prediction of microsatellite instability in colorectal cancer: A diagnostic study," *Lancet Oncol.*, vol. 22, no. 1, pp. 132–141, 2021.
[4] A. Raju, J. Yao, M. M. Haq, J. Jonnagaddala, and J. Huang, "Graph attention multi-instance learning for accurate colorectal cancer staging," in *Proc. 23rd Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2020, pp. 529–539.
[5] S. Wang et al., "RMDL: Recalibrated multi-instance deep learning for whole slide gastric image classification," *Med. Image Anal.*, vol. 58, 2019, Art. no. 101549.
[6] N. Lajara, J. L. Espinosa-Aranda, O. Deniz, and G. Bueno, "Optimum web viewer application for DICOM whole slide image visualization in anatomical pathology," *Comput. Methods Prog. Biomed.*, vol. 179, 2019, Art. no. 104983.
[7] X. Zhu, J. Yao, F. Zhu, and J. Huang, "WSISA: Making survival prediction from whole slide histopathological images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7234–7242.
[8] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.
[9] R. Yan et al., "Histopathological bladder cancer gene mutation prediction with hierarchical deep multiple-instance learning," *Med. Image Anal.*, vol. 87, 2023, Art. no. 102824.
[10] J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, and J. Huang, "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks," *Med. Image Anal.*, vol. 65, 2020, Art. no. 101789.
[11] G. Campanella et al., "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images," *Nature Med.*, vol. 25, no. 8, pp. 1301–1309, 2019.
[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[13] R. Li, J. Yao, X. Zhu, Y. Li, and J. Huang, "Graph CNN for survival analysis on whole slide pathological images," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2018, pp. 174–182.
[14] Z. Wang et al., "Hierarchical graph pathomic network for progression free survival prediction," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 227–237.
[15] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.
[16] J. Li, J. Chen, Y. Tang, C. Wang, B. A. Landman, and S. K. Zhou, "Transforming medical imaging with transformers? A comparative review of key properties, current progresses, and future perspectives," *Med. Image Anal.*, vol. 85, 2023, Art. no. 102762.
[17] F. Yu, X. Wang, R. Sali, and R. Li, "Single-cell heterogeneity-aware transformer-guided multiple instance learning for cancer aneuploidy prediction from whole slide histopathology images," *IEEE J. Biomed. Health Inform.*, early access, Mar. 28, 2023, doi: 10.1109/JBHI.2023.3262454.
[18] H. Li et al., "DT-MIL: Deformable transformer for multi-instance learning on histopathological image," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 206–216.
[19] Z. Shao et al., "TransMIL: Transformer based correlated multiple instance learning for whole slide image classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 2136–2147.
[20] X. Wang et al., "TransPath: Transformer-based self-supervised learning for histopathological image classification," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 186–195.

[21] Z. Gao et al., "Instance-based vision transformer for subtyping of papillary renal cell carcinoma in histopathological image," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 299–308.
[22] Y. Chen et al., "dMIL-transformer: Multiple instance learning via integrating morphological and spatial information for lymph node metastasis classification," *IEEE J. Biomed. Health Inform.*, early access, Jun. 13, 2023, doi: 10.1109/JBHI.2023.3285275.
[23] Z. Huang, H. Chai, R. Wang, H. Wang, Y. Yang, and H. Wu, "Integration of patch features through self-supervised learning and transformer for survival analysis on whole slide images," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 561–570.
[24] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.
[25] S. K. Zhou et al., "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," *Proc. IEEE*, vol. 109, no. 5, pp. 820–838, May 2021.
[26] H. Li, Q. Song, D. Gui, M. Wang, X. Min, and A. Li, "Reconstruction-assisted feature encoding network for histologic subtype classification of non-small cell lung cancer," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 9, pp. 4563–4574, Sep. 2022.
[27] H. Zhao et al., "SC2Net: A novel segmentation-based classification network for detection of COVID-19 in chest X-Ray images," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 8, pp. 4032–4043, Aug. 2022.
[28] X. Zhu, J. Yao, and J. Huang, "Deep convolutional neural network for survival analysis with pathological images," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2016, pp. 544–547.
[29] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," 2016, *arXiv:1606.05718*.
[30] N. Coudray et al., "Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning," *Nature Med.*, vol. 24, no. 10, pp. 1559–1567, 2018.
[31] R. Feng, X. Liu, J. Chen, D. Z. Chen, H. Gao, and J. Wu, "A deep learning approach for colonoscopy pathology WSI analysis: Accurate segmentation and classification," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 10, pp. 3700–3708, Oct. 2021.
[32] N. G. Laleh et al., "Benchmarking weakly-supervised deep learning pipelines for whole slide classification in computational pathology," *Med. Image Anal.*, vol. 79, 2022, Art. no. 102474.
[33] X. Liu et al., "Development of prognostic biomarkers by TMB-guided WSI analysis: A two-step approach," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 4, pp. 1780–1789, Apr. 2023.
[34] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, 2021, Art. no. 113679.
[35] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video summarization using deep neural networks: A survey," *Proc. IEEE*, vol. 109, no. 11, pp. 1838–1863, Nov. 2021.
[36] Z. Gao, J. Shi, and J. Wang, "GQ-GCN: Group quadratic graph convolutional network for classification of histopathological images," in *Proc. 24th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2021, pp. 121–131.
[37] D. Tellez, G. Litjens, J. van der Laak, and F. Ciompi, "Neural image compression for gigapixel histopathology image analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 567–578, Feb. 2021.
[38] R. Yan et al., "Breast cancer histopathological image classification using a hybrid deep neural network," *Methods*, vol. 173, pp. 52–60, 2020.
[39] M. Adnan, S. Kalra, and H. R. Tizhoosh, "Representation learning of histopathology images using graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 988–989.
[40] B. Li, Y. Li, and K. W. Eliceiri, "Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14318–14328.
[41] A. Vaswani et al., "Attention is all you need," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5998–6008, 2017.
[42] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technological Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.
[43] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[44] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
[45] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[46] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[47] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, "Neighborhood attention transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6185–6194.
[48] H. Chen et al., "GasHis-Transformer: A multi-scale visual transformer approach for gastric histopathology image classification," *Pattern Recognit.*, vol. 130, p. 108827, 2022.
[49] K. He et al., "Transformers in medical image analysis: A review," 2022, *arXiv:2202.12165*.
[50] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," 2019, *arXiv:1904.10509*.
[51] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue, and Z. Zhang, "Star-transformer," in *Proc. NAACL-HLT*, 2019, pp. 1315–1325.
[52] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.
[53] M. Zaheer et al., "Big bird: Transformers for longer sequences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17283–17297.
[54] L. Ericsson, H. Gouk, and T. M. Hospedales, "How well do self-supervised models transfer?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5414–5423.
[55] J.-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.
[56] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Emp. Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3730–3740.
[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[58] C. Wei, S. Kakade, and T. Ma, "The implicit and explicit regularization effects of dropout," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10181–10192.
[59] G. Zhao, J. Lin, Z. Zhang, X. Ren, Q. Su, and X. Sun, "Explicit sparse transformer: Concentrated attention through explicit selection," 2019, *arXiv:1912.11637*.
[60] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: Personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC Med. Res. Methodol.*, vol. 18, no. 1, pp. 1–12, 2018.
[61] C. Kandoth et al., "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, 2013.
[62] H. Steck, B. Krishnapuram, C. Dehing-Oberije, P. Lambin, and V. C. Raykar, "On ranking in survival analysis: Bounds on the concordance index," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 1209–1216, 2007.
[63] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[64] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.
[65] X. Han et al., "Construction of a human cell landscape at single-cell level," *Nature*, vol. 581, no. 7808, pp. 303–309, 2020.
[66] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.