Cell Graph Transformer for Nuclei Classification

Wei Lou[1, 2], **Guanbin Li**[3, 4], **Xiang Wan**[1], **Haofeng Li**[1*]

[1]Shenzhen Research Institute of Big Data, Shenzhen, China
[2]The Chinese University of Hong Kong, Shenzhen, China
[3]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[4]GuangDong Province Key Laboratory of Information Security Technology
weilou@link.cuhk.edu.cn, liguanbin@cuhk.edu.cn, {wanxiang, lhaof}@sribd.cn

Abstract

Nuclei classification is a critical step in computer-aided diagnosis with histopathology images. In the past, various methods have employed graph neural networks (GNN) to analyze cell graphs that model inter-cell relationships by considering nuclei as vertices. However, they are limited by the GNN mechanism that only passes messages among local nodes via fixed edges. To address the issue, we develop a cell graph transformer (CGT) that treats nodes and edges as input tokens to enable learnable adjacency and information exchange among all nodes. Nevertheless, training the transformer with a cell graph presents another challenge. Poorly initialized features can lead to noisy self-attention scores and inferior convergence, particularly when processing the cell graphs with numerous connections. Thus, we further propose a novel topology-aware pretraining method that leverages a graph convolutional network (GCN) to learn a feature extractor. The pre-trained features may suppress unreasonable correlations and hence ease the finetuning of CGT. Experimental results suggest that the proposed cell graph transformer with topology-aware pretraining significantly improves the nuclei classification results, and achieves the state-of-the-art performance. Code and models are available at https://github.com/lhaof/CGT

Introduction

Identifying cell types for histopathology image has emerged as a fundamental task in computational pathology (Krithiga and Geetha 2021; Amgad et al. 2022; Huang et al. 2023a). By effectively classifying nuclei, medical professionals gain crucial insights into the intricate cellular structures, which helps make decisions related to disease diagnosis (Lagree et al. 2021) and prognosis (Liu et al. 2022a). Thus, in this paper, we focus on inferring the types of cell nuclei in a histopathology image.

Deep learning (DL) based methods (Graham et al. 2019; Abousamra et al. 2021; Doan et al. 2022) have been widely applied to the nuclei classification task. Most of them employ convolutional neural networks (CNNs) to compute pixel-wise local features and fail to consider the macrostructure of nuclei distribution (Anand, Gadiya, and Sethi 2020; Javed et al. 2020). Another group of methods exploits the cell graph of a histopathology image, and has been studied for decades (Schnorrenberg et al. 1996; Hassan et al. 2022). A *Cell Graph* is a set of vertices and edges, where a vertex is a cell or nucleus and an edge is built between two neighboring cells. Recently, graph convolutional networks (GCNs) have been used to learn embeddings with cell graphs (Zhou et al. 2019; Zhao et al. 2020; Anklin et al. 2021; Hassan et al. 2022). These GCN-based solutions update the embedding of a nucleus by aggregating its adjacent nuclei. However, these GCN methods aggregate features along non-learnable edge connections that are fixed after building a cell graph, which limits the model capacity.

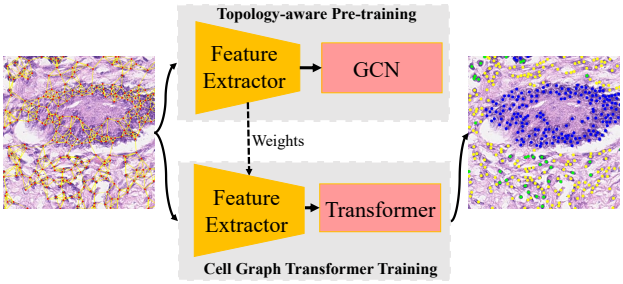

Figure 1: The idea of Topoloyg-aware Pre-training for Cell Graph Transformer. Simple initialization for the cell graph transformer fails to converge due to a large amount of unreasonable connections. The topology-aware pre-training can reduce the initial noise in features and boost the representation ability of the cell graph transformer.

To overcome the issue, we propose a Cell Graph Transformer (CGT) for the nuclei classification task, inspired by (Kreuzer et al. 2021; Ying et al. 2021; Kim et al. 2022). The proposed CGT takes both nuclei and edges as input tokens to compute any pairwise correlations among all tokens, which can capture long-range contexts in a more flexible way. CGT is a portable model that can identify cell types, based on any form of binary segmentation or detection results of nuclei. These results could be obtained by existing methods or manual labeling. In the CGT framework, we first compute the centroid coordinates of nuclei from the segmentation/detection result, then determine if two cells are connected by an edge according to their spatial distance, and build up the

*Haofeng Li is the corresponding author.

topology of a cell graph. To obtain visual features for nodes and edges, we develop a U-Net that outputs a pixel-wise feature map with an input image. To embed adjacency into the CGT, we propose a cell-graph tokenization method that integrates the visual and position embeddings of nodes and edges with two kinds of markers, which indicates the type and neighborhood of a token. Afterward, the CGT encoder, which is built of standard transformer layers, performs node-level classification to output the cell types.

Importantly, we observe that simple initialization of the feature extractor fails to train our proposed CGT model. It may be due to that the pairwise attentions can be computed between less correlated tokens, and result in noises especially when the representations are not well initialized. Therefore, we propose a novel topology-aware pretraining strategy that replaces our proposed CGT with a GCN to guide the learning of a feature extractor on the same nuclei classification task. The guiding GCN model only merges the embeddings of adjacent nodes defined by the fixed cell graph, which means less unreasonable correlations and makes convergence easier. The pre-trained feature extractor is supposed to synthesize structure-guided representations that benefit the training of the proposed CGT framework.

Overall, our contributions have three folds:

- A nuclei classification framework, Cell Graph Transformer, which benefits from non-local contexts by computing pairwise attentions among all nodes and edges;
- A topology-aware pretraining strategy that provides topology-guided feature learning for reducing the initial noise of the cell graph transformer;
- The proposed cell graph transformer significantly surpasses the state-of-the-art methods and our pretraining strategy also brings an improvement to the baseline.

Related Work

Nuclei Classification for Histopathology Images. Early solutions for nuclei classification involved the extraction of manually defined features which are fed into classifiers like SVM or AdaBoost (Liu, Mundra, and Rajapakse 2011; Sharma et al. 2015). However, the handcrafted features limit the representation capabilities of nuclei entities. Recently, the nuclei classification models usually infer cell types based on the CNNs for nucleus segmentation (Zhang et al. 2017; Basha et al. 2018; Lou et al. 2022, 2023b; Ma et al. 2023; Yu et al. 2023) or nucleus centroid detection (Abousamra et al. 2021; Huang et al. 2023b). Graham et al. (2019) propose a CNN of three branches, predicting nucleus types for the segmented nucleus instances. Doan et al. (2022) incorporated a weight map prediction technique to highlight challenging pixel samples for improved classification. However, these CNN-based approaches are limited by their local pixel-wise receptive field, and fail to capture instance-level contexts among cell nuclei. Therefore, we use a cell graph structure that describes the global relationship among nucleus instances.

Graph Models in Computational Pathology. Graph models have been used in computational pathology for decades. Demir, Gultekin, and Yener (2005) builds a graph by considering nuclei as nodes and binary connections as edges. A perceptron is utilized for the detection of inflammation in brain biopsy. Recently graph convolution networks (GCNs) have been used as learnable models for the graphs derived from histopathology images (Lou et al. 2023a). Some approaches (Zhou et al. 2019; Javed et al. 2020; Pati et al. 2022) classify whole slide images by defining nodes as nucleus instances, superpixels, or tissue patches. In these methods, the node embeddings are hand-crafted or learned features from pre-trained CNN models. NCCD (Hassan et al. 2022) has been proposed for GNN-based nucleus classification recently. However, the GNN-based methods aggregate features along non-learnable edges, which are fixed and limit the model capacity. Thus, we develop a cell graph transformer with learnable node connections and capture the long-range contexts more flexibly.

Transformers for Graph. Transformer models have emerged as crucial components in various domains such as neural language processing (Vaswani et al. 2017) and computer vision (Liu et al. 2021). Several existing approaches have incorporated transformers to handle graph structures in different manners. First, some methods employ Transformer layers as auxiliary modules within graph neural networks (Wu et al. 2021; Lin, Wang, and Liu 2021). Second, attention matrices are introduced into the message-passing mechanism (Dwivedi and Bresson 2020; Zheng et al. 2022). However, these approaches are constrained by the non-learnable edge connections in the graph structure and may suffer from the issue of excessive smoothing caused by the message-passing mechanism (Li, Han, and Wu 2018; Oono and Suzuki 2020). Recently, researchers have made progress in graph representation learning by employing pure Transformer architectures with learnable positional encodings (Kreuzer et al. 2021) or by utilizing sparse higher-order Transformers (Kim, Oh, and Hong 2021). In this paper, we propose a GCN-guided pretraining strategy that adapts visual features to a graph topology for better training a cell graph transformer on the nuclei classification task.

Methodology

In this section, we introduce the proposed Cell Graph Transformer framework, the proposed topology-aware pretraining strategy, and the training-inference scheme.

Cell Graph Transformer

We propose a cell graph transformer (CGT) framework to identify the category of each nucleus in histopathology images. To focus on the classification part, the CGT simply adopts an existing model to provide binary (foreground v.s. background) segmentation/detection results of nuclei. Since CGT performs nuclei classification based on the position coordinates of nuclei, it can adapt to various forms of segmentation/detection results. As Figure 2 shows, our proposed cell graph transformer has three parts: Cell Graph Construction, Cell Graph Tokenization (CGToken), and Transformer Encoder.

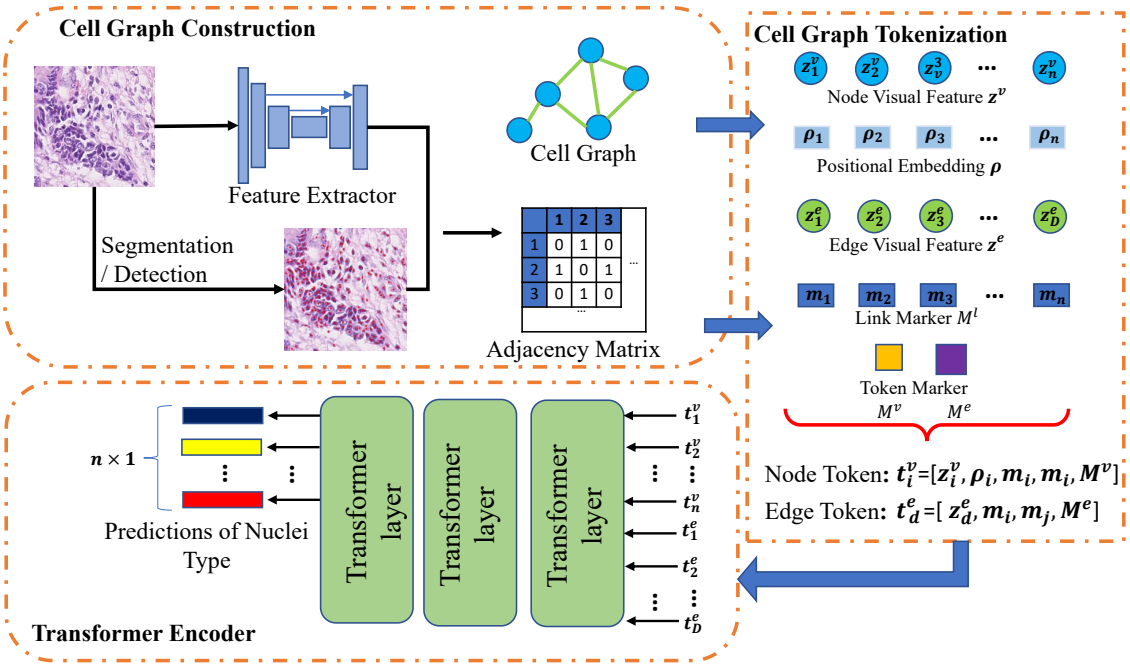

Figure 2: Overview of the Cell Graph Transformer (CGT) framework. The framework includes the construction and tokenization of cell graphs as well as a transformer encoder. Tokens are the input feature vectors of the transformer encoder. Adjacency is embedded into the tokens via link and token markers.

Cell Graph Construction. Building a cell graph requires two inputs: 1. Centroid coordinates of nuclei computed by the binary detector/segmentation tool; 2. A feature map f is obtained by the feature extractor in Figure 2. Given the centroids of n nuclei in an image, an undirected cell graph $G = (V, E)$ can be constructed by connecting each nucleus centroid to its nearest k nuclei. Therefore, the cell graph contains n nodes $V = \{v_1, ..., v_n\}$ and $D(= kn)$ edges $E = \{e_1, ..., e_D\}$. The connections of nodes are represented by a binary adjacency matrix $A \in R^{n \times n}$, in which $A_{i,j}$=1 if node v_i and v_j are connected.

To obtain the visual embeddings of nodes and edges, our approach incorporates a U-Net architecture that leverages an existing convolutional neural network (CNN) (Guo et al. 2023) as the encoder, and a feature pyramid network (FPN) (Lin et al. 2017) as the decoder. The U-Net is initialized via a novel pre-training strategy, which is introduced in the subsequent subsection. Given a histopathology image with dimensions $H \times W$, the U-Net takes the image as input and generates a feature map f of size $\frac{H}{4} \times \frac{W}{4} \times C$ from its second-to-last layer. To describe a nucleus node v_i, we first sample the visual embedding $z_i^v \in R^{1 \times C}$ located at the centroid coordinate of the nucleus from f. The vector z_i^v can be calculated through bilinear interpolation by using feature vectors at the four nearest integer coordinates on f. Besides, we inject spatial positional information by computing a positional embedding vector $\rho_i \in R^{1 \times C}$ using the Sinusoidal Position Encoding method (Vaswani et al. 2017). The node feature is defined as the concatenation of z_i^v and ρ_i. For an edge, the feature vector of its middle point is sampled from the feature map f using bilinear interpolation. The sampled vector is viewed as the edge visual embedding z_d^e.

Cell Graph Tokenization. Tokenization is to convert raw data into meaningful numerical representations called *Tokens* that can be well encoded by transformers. We introduce the Cell Graph Tokenization approach (CGToken), which aims to translate the constructed cell graph into a set of tokens that standard transformer models can effectively process. It is straightforward to regard the node and edge embeddings as $(n+D)$ independent inputs, but it overlooks the structural information contained within the graph. To exploit the topology structure, we utilize *Link Markers* and *Token Markers*. A cell graph transformer framework has two token markers denoted as M^v, M^e, which are two $1 \times C$ vectors and the learnable parameters of the framework. One is for node tokens, while the other is for edges. The two token markers are tuned in the training and fixed after training.

The link markers $M^l = \{m_1, m_2, ..., m_n\} \in R^{n \times c}$ are orthonormal vectors that imply the adjacency of each token. If v_i and v_j are linked by an edge, then $[m_i, m_j][m_i, m_i]^T = 1$ and $[m_i, m_j][m_j, m_j]^T = 1$, otherwise the dot production is 0. This mechanism makes the transformer assign more attention to the nodes connected in the cell graph. The link markers are calculated by the Laplacian eigendecomposition (Dwivedi et al. 2020) of the adjacency matrix A:

$$L = I - \Theta^{-\frac{1}{2}} A \Theta^{-\frac{1}{2}} = (M^l)^T \alpha M^l, \quad (1)$$

where Θ is the degree matrix of the graph, I is an identity matrix. α and M^l are the eigenvalues and eigenvectors, respectively. Then, we further define each node/edge token (t_i^v

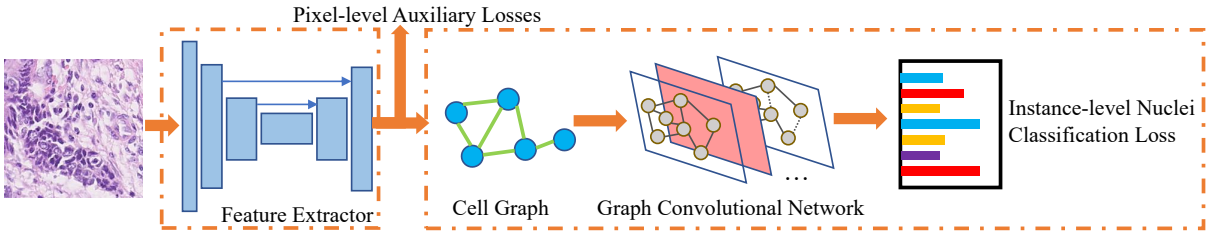

Figure 3: The proposed Topology-Aware Pretraining Strategy. It trains the feature extractor with an instance-level nuclei classification loss and two pixel-level auxiliary losses. The model weights of the trained feature extractor are then used to initialize the feature extractor in training the proposed cell graph transformer.

/ t_d^e) using node/edge visual feature, link and token makers:

$$t_i^v = [\sigma_1([z_i^v, \rho_i]), \sigma_3([m_i, m_i]), M^v], i \in \{1, \cdots, n\},$$
$$t_d^e = [\sigma_2(z_d^e), \sigma_3([m_i, m_j]), M^e], d \in \{1, \cdots, D\}, \quad (2)$$

where z_i^v and ρ_i are the visual and positional embeddings of the i^{th} node. z_d^e is the visual feature of the d^{th} edge that connects the i^{th} and j^{th} nodes. $[\cdot]$ denotes the concatenation operator. $\sigma_1, \sigma_2, \sigma_3$ are linear projection layers that convert the dimension of their inputs to the same dimension C. After computing Eq. (2), we obtain n node tokens and D edge tokens. Each token is a vector of size $1 \times 3C$.

Inference and Training Scheme

Given the $(n + D)$ tokens, a linear projection layer converts each of these tokens into a C-dimensional vector separately. The resulting $(n + D)$ vectors are then fed into the transformer encoder. We employ the standard transformer architecture (Vaswani et al. 2017) for each transformer layer in the encoder. Each layer is composed of a stack of multi-head self-attention layers and a feed-forward network. To classify the categories of n nodes, we only select the first n features $O \in R^{n \times C}$ from the output of the CGT encoder and send these features into the classification layer. The classification layer is built of a fully-connected (FC) layer and a Softmax function: $P = Softmax(\sigma(O))$.

Before training the CGT encoder, we pretrain the feature extractor using our proposed topology-aware strategy described in the next subsection. In the training stage, the feature extractor is also fine-tuned with the transformer encoder synchronously. The classification loss of each nucleus node has a cross-entropy term and a focal loss term, and is defined as:

$$\mathcal{L}(P, y) = -\sum_{b=1}^{B} y_b \log P_b - \sum_{b=1}^{B} \tau_b (1 - P_b)^{\gamma} y_b \log P_b, \quad (3)$$

where γ is a hyper-parameter set to 2, B is the number of categories, y is the true label, P is the prediction and τ_b is the loss weight computed as the reciprocal of the proportion of the b^{th} class in the training set.

Topology-Aware Pretraining Strategy

In the proposed CGT, we find that the initial visual features of nodes/edges play a crucial role in model training. At the early training stage, if visual features are not well initialized, computing correlations of all pairs of node/edge tokens may produce unreliable attention, bringing noise into the CGT training. Graph convolutional network (GCNs) and Transformers have their own strengths in modeling graph data. GCNs excel at capturing local structural information and propagating it across the graph. We consider that the message passing in GCNs is locally guided by fixed edges and is more robust at the start of training. Thus, we propose a pretraining strategy that employs a GCN to help learn the feature extractor in advance. After that, the cell graph transformer is initialized with the GCN-guided representations, which can help the transformer converge faster and improve the final classification performance.

As shown in Fig. 3, the proposed pretraining strategy trains a feature extractor with an instance-level nuclei classification loss and two auxiliary pixel-level losses (including the Dice and Cross-entropy losses). The auxiliary losses are for semantic segmentation, and the segmentation result is predicted from the last layer of the feature extractor. Note that these two segmentation losses aim at learning the feature extractor instead of producing nucleus masks. The segmentation mask of nuclei is obtained via an existing segmentation tool as shown in Fig. 2. To compute the instance-level loss in Fig. 3, the second-last layer of the feature extractor yields a feature map to build the node/edge features of a GCN (Li et al. 2021). The cell graph, the edges and their features are defined as the same as those in our proposed CGT. For nuclei segmentation and classification datasets (Gamper et al. 2020; Graham et al. 2021), we define the node features to exploit the segmentation masks, following Wei et al. (2023). After that, both the node and edge features are input to the GCN for node update. The enhanced node embeddings are fed into a linear classifier to predict nucleus types. The instance-level nuclei classification loss is the same as Eq. (3), by viewing P as the GCN prediction. In the pretraining, the feature extractor and the GCN are end-to-end tuned.

Experiments

Datasets. We utilize four nuclei classification datasets: PanNuke (Gamper et al. 2020), Lizard (Graham et al. 2021), NuCLS (Amgad et al. 2022), and BRCA-M2C (Abousamra et al. 2021). PanNuke, Lizard and NuCLS have the segmentation masks of nuclei, while BRCA-M2C only pro-

Method	PanNuke									NuCLS							
	AJI	PQ	F_d	F^i	F^c	F^d	F^{ep}	F^{ne}	$\boldsymbol{F_{avg}}$	AJI	PQ	F_d	F^t	F^{st}	F^s	F^o	$\boldsymbol{F_{avg}}$
MCSPat	-	-	0.786	0.484	0.473	0.220	0.612	0.629	0.514	-	-	0.658	0.488	0.267	0.581	0.035	0.343
Mask2former	0.616	0.666	0.792	0.400	0.426	0.289	0.668	0.617	0.480	0.229	0.331	0.432	0.367	0.098	0.521	0.000	0.247
SONNET	0.686	0.649	0.813	0.522	0.474	**0.367**	0.639	0.604	0.521	0.332	0.403	0.458	0.461	0.181	0.547	0.000	0.330
NCCD	-	-	0.800	**0.571**	0.525	0.354	0.660	0.588	0.539	-	-	-	-	-	-	-	-
Hover.	0.663	0.631	0.793	0.510	0.478	0.265	0.627	0.636	0.503	0.467	0.429	0.662	0.469	0.272	0.586	0.023	0.337
Ours+Hover.	0.663	0.631	0.793	0.527	**0.531**	0.358	**0.705**	**0.673**	**0.558**	0.467	0.429	0.662	**0.501**	**0.300**	**0.593**	**0.095**	**0.377**
Ours+GT	-	-	-	0.618	0.661	0.452	0.741	0.806	0.656	-	-	-	0.785	0.466	0.733	0.123	0.527

Table 1: Comparison with the state-of-the-art methods on PanNuke and NuCLS datasets. The best classification results are in bold. 'Hover.' and 'MCSPat' denotes Hover-net and MCSPatnet. 'Ours+Hover.' and 'Ours+GT' denote our CGT framework using the segmentation masks from a trained Hover-net model or ground truth.

Method	Lizard										Method	BRCA-M2C				
	AJI	PQ	F_d	F^{ne}	F^{ep}	F^l	F^p	F^e	F^c	$\boldsymbol{F_{avg}}$		F_d	F^i	F^{ep}	F^s	$\boldsymbol{F_{avg}}$
MCSPat.	-	-	0.705	0.110	0.604	0.457	0.228	0.210	0.478	0.347	DDOD	0.585	0.379	0.540	0.156	0.359
Mask2former	0.385	0.297	0.603	0.036	0.469	0.367	0.148	0.268	0.275	0.313	YOLOX	0.638	0.439	0.502	0.170	0.370
SONNET	0.434	0.447	0.597	0.197	0.610	0.322	0.328	0.402	0.421	0.380	ConvN.Uper.	0.785	0.423	0.636	0.353	0.471
NCCD	-	-	0.633	**0.378**	0.423	0.404	**0.471**	**0.461**	0.534	0.445	DINO	0.633	0.403	0.631	0.213	0.416
Hover.	0.463	0.460	0.732	0.221	0.693	0.447	0.369	0.387	0.493	0.435	MCSPat.	0.831	0.422	0.683	0.417	0.507
Ours+Hover.	0.463	0.460	0.732	0.302	**0.724**	**0.438**	0.434	0.416	**0.548**	**0.477**	Ours+MCSPat	0.831	**0.447**	**0.732**	**0.428**	**0.536**
Ours+GT	-	-	-	0.508	0.868	0.543	0.537	0.585	0.678	0.620	Ours+GT.	-	0.598	0.869	0.602	0.690

Table 2: Comparison with the state-of-the-art methods on Lizard and BRCA-M2C datasets. 'ConvNUper.' denotes the ConvNext-UperNet. Since the BRCA-M2C is a nuclei detection and classification benchmark, several nuclei detection and classification methods are utilized for comparison. The best classification results are in bold.

vides centroid annotations for nuclei detection and classification. The PanNuke dataset comprises 7901 images with a size of 256×256 from 19 organs, which includes the cell types of inflammatory, connective, dead, epithelial, and neoplastic. The Lizard benchmark consists of 291 large images with an average size of 1016×917, which is composed of six existing datasets: ConSeP (Graham et al. 2019), CRAG, GLAS (Sirinukunwattana et al. 2017), DigestPath, TCGA (Grossman et al. 2016), and PanNuke. Lizard contains the nucleus types of epithelial, lymphocyte, plasma, neutrophil, eosinophil, and connective. The NuCLS dataset has 1744 image patches that are grouped into four superclasses: tumor, stromal, sTILs, and other. The BRCA-M2C dataset includes 120 image patches collected from TCGA, has the cell types of inflammatory, epithelial, and stromal. The data split and more details are in the supplementary material.

Implementation details. The implementation is based on PyTorch (Paszke et al. 2017) and PyTorch Geometric library (Fey and Lenssen 2019). For the proposed CGT, the encoder and decoder of the feature extractor have four layers and three layers, respectively. The CGT encoder contains four transformer layers. For the pretraining strategy, the GCN is built of two GENConv (Li et al. 2020) layers. Our results are reported as the average result of training with three different random seeds. The dimensions of type markers and link markers are 64 and 16. The number of edges of each node is 4. The pretraining strategy and the training of CGT are run for 150 and 50 epochs, respectively, with the Adam optimizer in an NVIDIA A-100 GPU. The initial learning rates for pretraining and training are 10^{-4} and 10^{-5}, respectively. The overall training time is 2 days for each dataset.

Metrics. We utilize F-score (Graham et al. 2019) for evaluating classification performance. F^i, F^c, F^d, F^{ep}, F^{ne}, F^t, F^{st}, F^s, F^o, F^n, F^l, F^p, F^e denote the class-wise F-score for inflammatory, connective, dead, epithelial, neoplastic, tumor, stromal, sTIL, other, neutrophil, lymphocyte, plasma, eosinophil, respectively. F_{avg} denotes the average F-score for all classes in the same dataset. For evaluating segmentation and detection, we adopt Aggregated Jaccard Index (AJI) (Mahmood et al. 2019), Panoptic Quality (PQ) (Kirillov et al. 2019), and Detection Quality (F_d) (Graham et al. 2019).

Comparison with the State-of-the-art Methods

For PanNuke, Lizard and NuCLS datasets, the proposed CGT is compared with existing methods: Hover-net (Graham et al. 2019), MCSPatnet (Abousamra et al. 2021), SONNET (Doan et al. 2022), Mask2former (Cheng et al. 2022), NCCD (Hassan et al. 2022). Among them, Hover-net, SONNET and Mask2former are nuclei segmentation and classification methods, MCSPatnet is a nuclei detection and classification method and NCCD is a pure nuclei classification method. For BRCA-M2C dataset, we compare the proposed method with nuclei detection and classification methods: DDOD (Chen et al. 2021), YOLOX (Ge et al. 2021), ConvNext-UperNet (Liu et al. 2022b), MCSPatnet (Abousamra et al. 2021) and DINO (Zhang et al. 2022). In Table 1 & 2, 'Ours+Hover-net' or 'Ours+MCSpat' indicates that our CGT utilizes Hover-net or MCSPatNet to generate nuclei segmentation or detection results, without using

Models	Initializing Feature Extractor	Classifier	F^i	F^c	F^d	F^{ep}	F^{ne}	F_{avg}
M1	UNet ImageNet pretrained	Linear	0.510	0.463	0.065	0.668	0.000	0.341
M2	UNet ImageNet pretrained	Transformer	0.456	0.381	0.166	0.601	0.614	0.444
M3	UNet ImageNet pretrained	GCN	0.513	0.506	0.263	0.674	0.633	0.518
M4	UNet ImageNet pretrained	CGToken+Transformer	0.435	0.104	0.057	0.526	0.602	0.344
M5	UNet Linear pretrained	CGToken+Transformer	0.518	0.472	0.138	0.674	0.237	0.420
M6	UNet Transformer pretrained	CGToken+Transformer	0.484	0.438	0.237	0.634	0.626	0.484
M7	UNet GCN pretrained	Transformer	0.525	0.484	0.228	0.672	0.655	0.512
M8 (Ours)	UNet GCN pretrained	CGToken+Transformer	**0.527**	**0.531**	**0.358**	**0.705**	**0.673**	**0.558**

Table 3: Ablation study on PanNuke dataset. 'CGToken+Transformer' is the proposed classifier in our CGT framework. All the results are based on the official data split of the PanNuke dataset. The best results are in bold.

Method	#Para. (M)	Infer Time (s)	Model Size (Mb)
Hover-net	33.60	1799	144
Ours	37.43	447	465

Table 4: Computational efficiency on whole slide images. Inference time is measured as the average time of inferring ten whole slide images.

the predictions of cell types. The numerical results of SONNET and NCCD are collected from their papers. As Table 1 shows, our proposed method 'Ours+Hover-net' outperforms the second best models by 1.9%, 3.4% and 3.2% in F_{avg} on PanNuke, NuCLS and Lizard, respectively. 'Ours+MCSPat.' surpasses the second-best model by 2.9% in F_{avg} on the BRCA-M2C dataset. Figure 4 presents a visual comparison between our proposed CGT and Hover-net on two datasets. Both methods employ the same segmentation masks, but our method shows more accurate classification results of nuclei. More visual results can be found in the appendix.

Effectiveness and Generalization of CGT. Note that the segmentation tools used in our CGT can also produce classification results. Thus, the CGT is compared with them to show its strength. Comparing 'Ours+Hover' with Hover-net suggests that the CGT brings a significant improvement of 1.9%-3.4% in the average F-score on three benchmarks. Importantly, on the BRCA-M2C dataset, the proposed CGT also boosts MCSPatNet by 2.9% in F_{avg} to achieve the state-of-the-art performance. The improvements with two different segmentation/detection tools verify the generalization of our CGT framework.

In Table 1 & 2, 'Ours+GT' means that our CGT accesses the ground truth of binary segmentation in the testing. It shows that with more accurate nuclei centroids, our CGT can produce better classification results. We claim that the proposed cell graph transformer is a flexible framework that can infer cell types with various segmentation/detection models or manual annotations.

Computational efficiency of CGT. Table 4 displays the efficiency comparison between our proposed CGT and Hover-net. To evaluate the feasibility of real-world applications of CGT, we assessed the average parameter count (#Para), inference time (Infer Time), and model size on ten whole slide images (WSIs). These WSIs have an average size of 36210×71309. 'Ours' in Table 4 measures our method without including the segmentation tool. Our method increases 400+ MB storage and 25% inference time when cooperating with existing segmentation methods, which is acceptable considering the low cost of the hard disk and the significant improvement of performance.

Ablation Study

In Table 3, we assess the strengths of CGT and the proposed Topology-Aware Pretraining (TAP) strategy. In the testing stage, all models adopt the binary segmentation masks generated by the same Hover-net. 'UNet' denotes the feature extractor in CGT. 'UNet ImageNet pretrained' is to initial the UNet with the ImageNet-1K pretrained weights. 'Linear', 'Transformer', 'GCN' and 'CGToken+Transformer' denote four classifiers: a linear embedding layer, a vanilla transformer without graph structure, a graph convolutional network and our proposed cell graph tokenization with a transformer encoder, respectively. For example, UNet Linear/Transformer/GCN pretrained means using Linear/Transformer/GCN as the classifier to pretrain the UNet for initialization. Our method is denoted as M8 where 'UNet GCN pretrained' represents the TAP strategy.

Effectiveness of the proposed pretraining. To validate the proposed pretraining strategy, we compare M8 with M4-M6 and find that M8 using the TAP strategy significantly outperforms M4-M6 by 21%, 13% and 7.4% in F_{avg}, respectively. The above results suggest that the TAP strategy using GCN is more effective than the simple ImageNet-pretraining, the pretraining guided by a linear layer and a vanilla transformer. We claim that it is because the TAP strategy makes the visual features aware of the graph connections and better adapt to our proposed CGT classifier.

Effectiveness of the CGT classifier. Comparing M8 to M7 suggests that our proposed classifier built of cell graph tokenization and transformer encoder surpasses the vanilla transformer classifier without graph modeling by 4.6% in F_{avg}. Comparing M5 to M1 and M8 to M3 shows that our proposed CGT classifier can outperform the linear and the GCN classifier by 7.9% and 4% in F_{avg}. Besides, M1-M3 can be viewed as the combinations of the existing initialization and classifiers, and our overall method M8 exceeds these solutions by 4%-21% in F_{avg}.

Investigation of Hyper-parameters. In Figure 5, we study

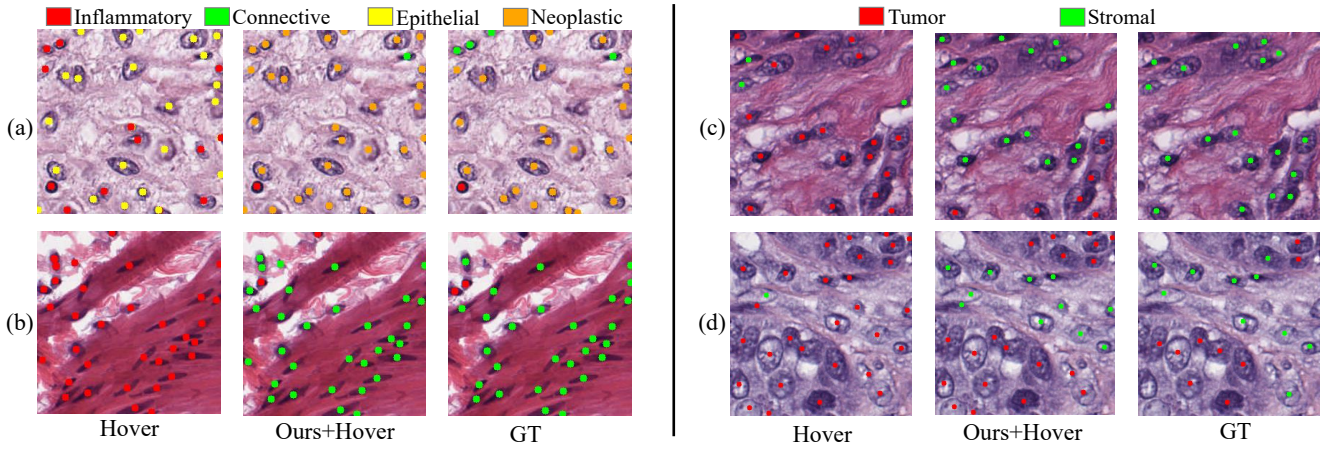

Figure 4: Visual comparison of the proposed CGT with Hover-net on PanNuKe (left) and NuCLS (right) datasets. GT denotes the ground truth.

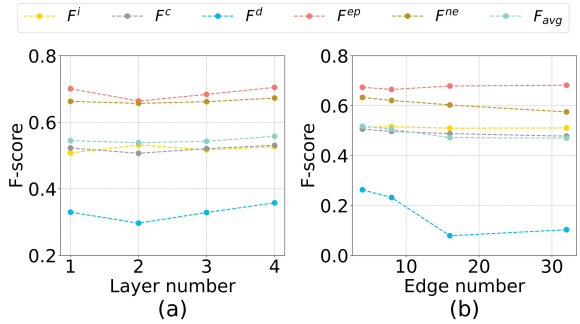

Figure 5: Analysis of different choices for layer number ($L \in \{1, 2, 3, 4\}$) in CGT and edge number ($E \in \{4, 8, 16, 32\}$) in GCN (the topology-aware pretraining) on PanNuke dataset.

the number of transformer layers L of the CGT encoder on PanNuke dataset. If we set L from 1 to 4, the average F-score first decrease by 0.6% and then increases by 0.4% and 1.5%. The results indicate that the performance is improved slightly with increasing transformer layers. Due to the GPU memory limitation, we do not test for larger layer numbers. The idea of our pretraining is that too dense connections with inferior features could result in unreasonable correlations and message passing, which affects the CGT training. In contrast, the GCN using a sparse graph is more robust. To verify the above idea, we use a denser graph for GCN-based pretraining by increasing edge number E. In Figure 5, as E increases from 4 to 8 and 32, the average F-score of GCN does drop by 1.2% and 4.8%, which validates our assumption.

Discussion

CGT vs. Transformer. The proposed CGT differs from vanilla transformers that compute correlations and pass messages between each pair of nodes equally. In contrast, the CGT defines edge features to describe pathological microenvironment and exploits link & token markers to learn connections which emphasizes the attention between relevant cells. In Table 3, the CGT (M8) outperforms the vanilla transformer (M7/M2) by 4.6%-11% in F_{avg}, which indicates that the proposed CGT better models the cells and their interactions in pathological images than the vanilla transformers.

GCN pretraining vs. Transformer pretraining. We discuss why the pretraining with GCN as classifier (M8 in Table 3) is better than the one with vanilla transformer (M6). The GCN-pretraining adopts a sparse graph where the well-defined connections could guide the reasonable propagation of information. During the GCN-pretraining, the feature extractor is tuned by the gradients that are computed based on the well-defined edges and can adapt to the topology of cell graphs. However, in the transformer-pretraining, the gradients passed through the feature extractor are calculated from any pairs of nodes, even those that are irrelevant. Thus, the gradients in transformer-pretraining are more noisy and unreliable than those in GCN-pretraining, at the start of training.

Conclusion

In this paper, a cell graph transformer (CGT) framework is proposed for identifying cell types with detected nucleus centroids. Our method embraces the transformer as a cell graph learner to fully exploit contexts and learn topological features. Both cell nodes and edges are viewed as input tokens to capture long-range correlations. The graph structure is embedded into the transformer encoder via link markers and token markers. Furthermore, we develop a novel topology-aware pretraining strategy that employs the robust local message-passing mechanism of graph convolutional networks to help pretrain the feature extractor of CGT. The experimental results display that the CGT model achieves the state-of-the-art nuclei classification performance on existing benchmarks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NO. 62102267, NO. 62322608), in part by the Natural Science Foundation of Guangdong Province of China (2023A1515011464), in part by the Shenzhen Science and Technology Program JCYJ20220818103001002), and in part by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen.

References

Abousamra, S.; Belinsky, D.; Van Arnam, J.; Allard, F.; Yee, E.; Gupta, R.; Kurc, T.; Samaras, D.; Saltz, J.; and Chen, C. 2021. Multi-class cell detection using spatial context representation. In *ICCV*, 4005–4014.

Amgad, M.; Atteya, L. A.; Hussein, H.; Mohammed, K. H.; Hafiz, E.; Elsebaie, M. A.; Alhusseiny, A. M.; AlMoslemany, M. A.; Elmatboly, A. M.; Pappalardo, P. A.; et al. 2022. NuCLS: A scalable crowdsourcing approach and dataset for nucleus classification and segmentation in breast cancer. *GigaScience*, 11.

Anand, D.; Gadiya, S.; and Sethi, A. 2020. Histographs: graphs in histopathology. In *Medical Imaging 2020: Digital Pathology*, volume 11320, 150–155. SPIE.

Anklin, V.; Pati, P.; Jaume, G.; Bozorgtabar, B.; Foncubierta-Rodriguez, A.; Thiran, J.-P.; Sibony, M.; Gabrani, M.; and Goksel, O. 2021. Learning whole-slide segmentation from inexact and incomplete labels using tissue graphs. In *MICCAI*, 636–646. Springer.

Basha, S. S.; Ghosh, S.; Babu, K. K.; Dubey, S. R.; Pulabaigari, V.; and Mukherjee, S. 2018. Rccnet: An efficient convolutional neural network for histological routine colon cancer nuclei classification. In *ICARCV*, 1222–1227. IEEE.

Chen, Z.; Yang, C.; Li, Q.; Zhao, F.; Zha, Z.-J.; and Wu, F. 2021. Disentangle your dense object detector. In *ACM Multimedia*, 4939–4948.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 1290–1299.

Demir, C.; Gultekin, S. H.; and Yener, B. 2005. Augmented cell-graphs for automated cancer diagnosis. *Bioinformatics*, 21(suppl_2): ii7–ii12.

Doan, T. N. N.; Song, B.; Le Vuong, T. T.; Kim, K.; and Kwak, J. T. 2022. SONNET: A self-guided ordinal regression neural network for segmentation and classification of nuclei in large-scale multi-tissue histology images. *IEEE JBHI*.

Dwivedi, V. P.; and Bresson, X. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.

Dwivedi, V. P.; Joshi, C. K.; Laurent, T.; Bengio, Y.; and Bresson, X. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *ICLR Workshop*.

Gamper, J.; Koohbanani, N. A.; Benes, K.; Graham, S.; Jahanifar, M.; Khurram, S. A.; Azam, A.; Hewitt, K.; and Rajpoot, N. 2020. Pannuke dataset extension, insights and baselines. *arXiv preprint arXiv:2003.10778*.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Graham, S.; Jahanifar, M.; Azam, A.; Nimir, M.; Tsang, Y.-W.; Dodd, K.; Hero, E.; Sahota, H.; Tank, A.; Benes, K.; et al. 2021. Lizard: A large-scale dataset for colonic nuclear instance segmentation and classification. In *ICCV Workshops*, 684–693.

Graham, S.; Vu, Q. D.; Raza, S. E. A.; Azam, A.; Tsang, Y. W.; Kwak, J. T.; and Rajpoot, N. 2019. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *MIA*, 58: 101563.

Grossman, R. L.; Heath, A. P.; Ferretti, V.; Varmus, H. E.; Lowy, D. R.; Kibbe, W. A.; and Staudt, L. M. 2016. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12): 1109–1112.

Guo, M.-H.; Lu, C.-Z.; Liu, Z.-N.; Cheng, M.-M.; and Hu, S.-M. 2023. Visual attention network. *Computational Visual Media*, 1–20.

Hassan, T.; Javed, S.; Mahmood, A.; Qaiser, T.; Werghi, N.; and Rajpoot, N. 2022. Nucleus Classification in Histology Images Using Message Passing Network. *MIA*, 102480.

Huang, J.; Li, H.; Sun, W.; Wan, X.; and Li, G. 2023a. Prompt-based grouping transformer for nucleus detection and classification. In *MICCAI*, 569–579. Springer.

Huang, J.; Li, H.; Wan, X.; and Li, G. 2023b. Affine-Consistent Transformer for Multi-Class Cell Nuclei Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21384–21393.

Javed, S.; Mahmood, A.; Fraz, M. M.; Koohbanani, N. A.; Benes, K.; Tsang, Y.-W.; Hewitt, K.; Epstein, D.; Snead, D.; and Rajpoot, N. 2020. Cellular community detection for tissue phenotyping in colorectal cancer histology images. *MIA*, 63: 101696.

Kim, J.; Nguyen, D. T.; Min, S.; Cho, S.; Lee, M.; Lee, H.; and Hong, S. 2022. Pure Transformers are Powerful Graph Learners. *NeurIPS*.

Kim, J.; Oh, S.; and Hong, S. 2021. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *NeurIPS*, 34: 28016–28028.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *CVPR*, 9404–9413.

Kreuzer, D.; Beaini, D.; Hamilton, W.; Létourneau, V.; and Tossou, P. 2021. Rethinking graph transformers with spectral attention. *NeurIPS*, 34: 21618–21629.

Krithiga, R.; and Geetha, P. 2021. Breast cancer detection, segmentation and classification on histopathology images analysis: a systematic review. *Archives of Computational Methods in Engineering*, 28: 2607–2619.

Lagree, A.; Mohebpour, M.; Meti, N.; Saednia, K.; Lu, F.-I.; Slodkowska, E.; Gandhi, S.; Rakovitch, E.; Shenfield, A.; Sadeghi-Naini, A.; et al. 2021. A review and comparison

of breast tumor cell nuclei segmentation performances using deep convolutional neural networks. *Scientific Reports*, 11(1): 8025.

Li, G.; Müller, M.; Qian, G.; Perez, I. C. D.; Abualshour, A.; Thabet, A. K.; and Ghanem, B. 2021. Deepgcns: Making gcns go as deep as cnns. *TPAMI*, 6923 – 6939.

Li, G.; Xiong, C.; Thabet, A.; and Ghanem, B. 2020. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, volume 32.

Lin, K.; Wang, L.; and Liu, Z. 2021. Mesh graphormer. In *ICCV*, 12939–12948.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Liu, S.; Mundra, P. A.; and Rajapakse, J. C. 2011. Features for cells and nuclei classification. In *EMBC*, 6601–6604.

Liu, Y.; Jia, Y.; Hou, C.; Li, N.; Zhang, N.; Yan, X.; Yang, L.; Guo, Y.; Chen, H.; Li, J.; et al. 2022a. Pathological prognosis classification of patients with neuroblastoma using computational pathology analysis. *CBM*, 149: 105980.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022b. A convnet for the 2020s. In *CVPR*, 11976–11986.

Lou, W.; Li, H.; Li, G.; Han, X.; and Wan, X. 2022. Which pixel to annotate: a label-efficient nuclei segmentation framework. *IEEE Transactions on Medical Imaging*, 42(4): 947–958.

Lou, W.; Wan, X.; Li, G.; Lou, X.; Li, C.; Gao, F.; and Li, H. 2023a. Structure Embedded Nucleus Classification for Histopathology Images. *arXiv preprint arXiv:2302.11416*.

Lou, W.; Yu, X.; Liu, C.; Wan, X.; Li, G.; Liu, S.; and Li, H. 2023b. Multi-stream Cell Segmentation with Low-level Cues for Multi-modality Images. In *Competitions in Neural Information Processing Systems*, 1–10. PMLR.

Ma, J.; Xie, R.; Ayyadhury, S.; Ge, C.; Gupta, A.; Gupta, R.; Gu, S.; Zhang, Y.; Lee, G.; Kim, J.; et al. 2023. The Multi-modality Cell Segmentation Challenge: Towards Universal Solutions. *arXiv preprint arXiv:2308.05864*.

Mahmood, F.; Borders, D.; Chen, R. J.; McKay, G. N.; Salimian, K. J.; Baras, A.; and Durr, N. J. 2019. Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE TMI*, 39(11): 3257–3267.

Oono, K.; and Suzuki, T. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *ICLR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. *NIPS Workshop*.

Pati, P.; Jaume, G.; Foncubierta-Rodríguez, A.; Feroce, F.; Anniciello, A. M.; Scognamiglio, G.; Brancati, N.; Fiche, M.; Dubruc, E.; Riccio, D.; et al. 2022. Hierarchical graph representations in digital pathology. *MIA*, 75: 102264.

Schnorrenberg, F.; Pattichis, C. S.; Schizas, C. N.; Kyriacou, K.; and Vassiliou, M. 1996. Computer-aided classification of breast cancer nuclei. *Technology and Health Care*, 4(2): 147–161.

Sharma, H.; Zerbe, N.; Heim, D.; Wienert, S.; Behrens, H.-M.; Hellwich, O.; and Hufnagl, P. 2015. A multi-resolution approach for combining visual information using nuclei segmentation and classification in histopathological images. In *VISAPP (3)*, 37–46.

Sirinukunwattana, K.; Pluim, J. P.; Chen, H.; Qi, X.; Heng, P.-A.; Guo, Y. B.; Wang, L. Y.; Matuszewski, B. J.; Bruni, E.; Sanchez, U.; et al. 2017. Gland segmentation in colon histology images: The glas challenge contest. *MIA*, 35: 489–502.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NIPS*, 30.

Wei, L.; Xiang, W.; Guanbin, L.; Xiaoying, L.; Chenghang, L.; Feng, G.; and Li, H. 2023. Structure Embedded Nucleus Classification for Histopathology Images. *arXiv preprint arXiv:2302.11416*.

Wu, Z.; Jain, P.; Wright, M.; Mirhoseini, A.; Gonzalez, J. E.; and Stoica, I. 2021. Representing long-range context for graph neural networks with global attention. *NeurIPS*, 34: 13266–13279.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do transformers really perform badly for graph representation? *NeurIPS*, 34: 28877–28888.

Yu, X.; Li, G.; Lou, W.; Liu, S.; Wan, X.; Chen, Y.; and Li, H. 2023. Diffusion-based data augmentation for nuclei image segmentation. In *MICCAI*, 592–602. Springer.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.; and Shum, H.-Y. 2022. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *ICLR*.

Zhang, L.; Lu, L.; Nogues, I.; Summers, R. M.; Liu, S.; and Yao, J. 2017. DeepPap: deep convolutional networks for cervical cell classification. *IEEE JBHI*, 21(6): 1633–1643.

Zhao, Y.; Yang, F.; Fang, Y.; Liu, H.; Zhou, N.; Zhang, J.; Sun, J.; Yang, S.; Menze, B.; Fan, X.; et al. 2020. Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *CVPR*, 4837–4846.

Zheng, Y.; Gindra, R. H.; Green, E. J.; Burks, E. J.; Betke, M.; Beane, J. E.; and Kolachalama, V. B. 2022. A graph-transformer for whole slide image classification. *IEEE TMI*, 41(11): 3003–3015.

Zhou, Y.; Graham, S.; Alemi Koohbanani, N.; Shaban, M.; Heng, P.-A.; and Rajpoot, N. 2019. Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images. In *ICCV Workshops*.