

Histopathology Whole Slide Image Analysis with Heterogeneous Graph Representation Learning

Tsai Hor Chan^{1,*}, Fernando Julio Cendra^{1,2*}, Lan Ma², Guosheng Yin^{1,3}, Lequan Yu¹

¹Department of Statistics and Actuarial Science, The University of Hong Kong

²TCL Corporate Research Hong Kong

³Department of Mathematics, Imperial College London

{hchanth, fcendra}@connect.hku.hk, rubyma@tcl.com, guosheng.yin@imperial.ac.uk, lqyu@hku.hk

Abstract

Graph-based methods have been extensively applied to whole slide histopathology image (WSI) analysis due to the advantage of modeling the spatial relationships among different entities. However, most of the existing methods focus on modeling WSIs with homogeneous graphs (e.g., with homogeneous node type). Despite their successes, these works are incapable of mining the complex structural relations between biological entities (e.g., the diverse interaction among different cell types) in the WSI. We propose a novel heterogeneous graph-based framework to leverage the inter-relationships among different types of nuclei for WSI analysis. Specifically, we formulate the WSI as a heterogeneous graph with “nucleus-type” attribute to each node and a semantic similarity attribute to each edge. We then present a new heterogeneous-graph edge attribute transformer (HEAT) to take advantage of the edge and node heterogeneity during message aggregating. Further, we design a new pseudo-label-based semantic-consistent pooling mechanism to obtain graph-level features, which can mitigate the over-parameterization issue of conventional cluster-based pooling. Additionally, observing the limitations of existing association-based localization methods, we propose a causal-driven approach attributing the contribution of each node to improve the interpretability of our framework. Extensive experiments on three public TCGA benchmark datasets demonstrate that our framework outperforms the state-of-the-art methods with considerable margins on various tasks. Our codes are available at <https://github.com/HKU-MedAI/WSI-HGNN>.

1. Introduction

Histopathology slides provide rich information on diagnosis and treatment planning for many cancer diseases. The

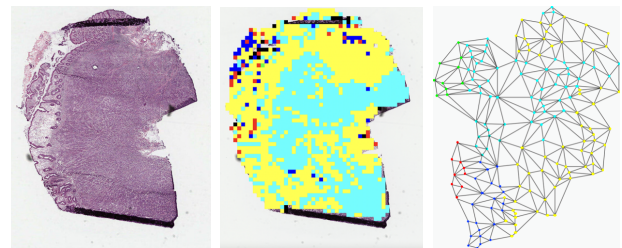


Figure 1. **Left:** Input WSI. **Middle:** A WSI with selected patches and associated node types. (Black - no label; cyan - neoplastic; red - inflammatory; blue - connective; yellow - dead; green - non-neoplastic epithelial). **Right:** Constructed heterogeneous graph with different types of nodes and edge attributes (Illustrative).

recent technological advancements in tissue digital scanners facilitate the development of whole slide histopathology image (WSI) analysis. However, traversing through the WSI with diverse magnifications is time-consuming and tedious for pathologists due to the large-scale nature of the WSI (e.g., its typical size is $60,000 \times 60,000$ pixels). Hence deep learning techniques play an important role as they introduce accurate and automated analysis of WSIs, which can significantly relieve the workload of pathologists.

Since it is difficult to fit the complete WSI into the memory, most of the works adopt multiple instance learning (MIL) to divide the WSI into instances and then aggregate them for WSI analysis. However, these methods operate on bags of instances that do not emphasize the inter-relationships between these instances. Recently, the emergence of graph neural networks (GNNs) has made large progress in representing the spatial relationships between instances. As a result, there are many attempts to represent the WSIs as graphs of instances. Figure 1 presents an example of a graph constructed from WSI. Unlike convolutional neural networks (CNNs) that aggregate features based on locality in the Euclidean space, GNNs focus on locality on graph topology, which offers more flexibility in analyzing the deep connections between features in the image data be-

*The first two authors contributed equally to this work.

yond the spatial locality [1]. For example, GNNs are able to learn relational information and distinguish cells based on their apposition to tumor cells, or normal stroma (i.e., cells which are tumor-infiltrating lymphocytes or from an adjacency inflammatory response), which are important for prognosis [5,27].

However, existing paradigms on graph-based WSI analysis focus on representing the WSI with a homogeneous graph structure and then predicting the response via vanilla GNNs with cluster-based pooling (i.e., based on similarities of node embeddings). Despite their successes, these methods suffer from several drawbacks: (i) GNNs on homogeneous graphs focus on aggregating direct relational information from neighboring nodes, where the complex relational information of the graphs is often neglected. (ii) For different graphs, the clusters defined by similarities between node embeddings have inconsistent meanings. This introduces a large degree of freedom in parameters and leads to over-parameterization issue [2]. Therefore, GNNs tend to easily overfit due to a lack of identifiability [14].

In view of these limitations, we propose a novel framework for WSI analysis, which leverages a heterogeneous graph to learn the inter-relationships among different types of nodes and edges. The heterogeneous graph introduces a “nucleus-type” attribute to each node, which can serve as an effective data structure for modeling the structural interactions among the nuclei in the WSI. To tackle the aggregation process in the heterogeneous graph, we propose a novel heterogeneous-graph edge attribute transformer (HEAT) architecture which can take advantage of the edge and node heterogeneity. Thus, the diverse structural relations among different biological entities in the WSI can be incorporated to guide the GNN for more accurate prediction. Further, to obtain the graph-level representations for slide-level prediction, we propose a semantic-consistent pooling mechanism — pseudo-label (PL) pooling, which pools node features to graph level based on clusters with a fixed definition (i.e., nucleus type). The proposed PL pooling can regularize the graph pooling process by distilling the context knowledge (i.e., pathological knowledge) from a pretrained model to alleviate the over-parameterization issue [2]. Additionally, we propose a Granger causality [13] based localization method to identify the potential regions of interest with clinical relevance to provide more insights to pathologists and promote the clinical usability of our approach.

We extensively evaluate our method on three TCGA public benchmark datasets, including colon adenocarcinoma cancer (COAD) and breast invasive carcinoma (BRCA) datasets from the TCGA project [35] and the Camelyon 16 dataset [3], and compare to various latest state-of-the-art (SOTA) methods. Our method outperforms the competitors on cancer staging, cancer classification, cancer typing, and localization tasks.

2. Related Works

Multiple Instance Learning on WSIs. Existing WSI analysis approaches generally adopt MIL [5,7,12,26,30,33,41], which first divide the WSI into fixed-size patches and then compress the information of these patches into low-dimensional vectors. Conventional methods aggregate bags of instances to learn WSI-level features for final predictions. Tellez *et al.* [30] compress the WSI-level image into embedding vectors and use a standard CNN to perform patch-level and WSI-level cancer classification. These CNN-based methods analyze local areas in the Euclidean space on fixed connectivity (i.e., fixed-size kernels), limiting the performance beyond the spatial locality. Graph-based methods [5,15,41] have recently been proposed, which model the interactions between instances via graphs. Their capability of modeling instances based on graph topology provides more flexibility to analyze complex structures of WSIs. Chen *et al.* [5] propose patch-GCN, a method of modeling WSI with homogeneous graphs, and regress survival data with a graph convolutional neural network (GCN) [36]. Zheng *et al.* [41] propose a graph-based MIL method using graph transformer networks [40]. In spite of their power, most of these WSI methods use homogeneous graphs, which limits the information mined from WSIs. A recent method [15] is proposed to model WSIs with heterogeneous graphs, where the heterogeneity in each patch is introduced by different resolution levels. However, it only considers the resolution level heterogeneity of patches, with insufficient ability to model the complex contextual interaction between patches in the same resolution level.

Graph Neural Networks. Although the SOTA GNNs have shown great successes in many problem domains [16,19,20], they are mostly focused on homogeneous graphs [32,36,37,40,42]. These architectures extract the locality information on the graph topology and learn the graph representations by performing aggregation on neighboring nodes. However, the potential heterogeneity in nodes and edges is not incorporated by these homogeneous GNN algorithms, and therefore their capability in mining the structural information is limited. Several works attempt to address the heterogeneity in their architectural designs [16,28,34] and assume that the relation type is finite and discrete. However, when modeling images with graphs, the heterogeneity in relations is typically continuous (e.g., the similarity between nodes) or high-dimensional. Although there are several attempts [5,10] to extend SOTA GNNs [32,36] to incorporate edge attributes, their works are limited to homogeneous graphs.

Graph Pooling. Graph pooling aims to aggregate node-level features to obtain graph-level features. Conventional methods [36] directly take the average of node-level features to extract graph-level features, which tends to over-

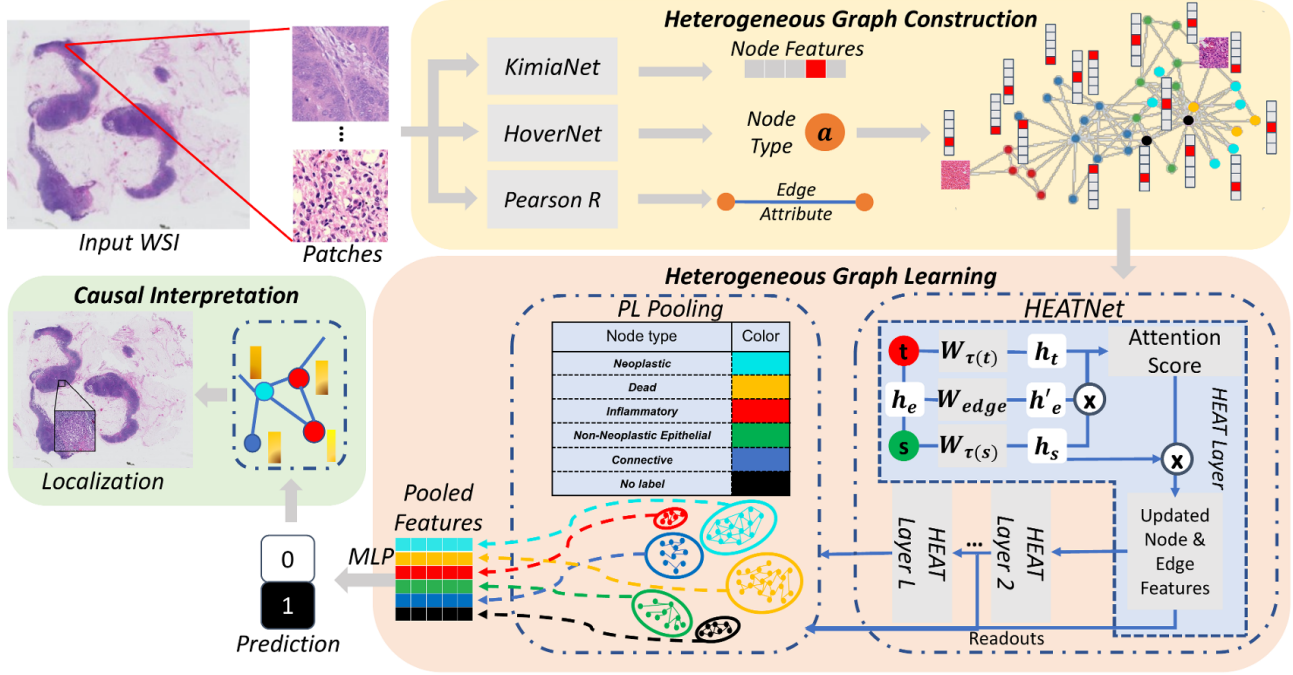


Figure 2. The paradigm of our proposed heterogeneous graph-based WSI analysis framework, which includes heterogeneous graph construction, heterogeneous-graph edge attribute transformer (HEAT) for structural information aggregation, pseudo-label-based (PL) graph pooling for slide-level prediction and casual-driven localization.

smooth the signals of the nodes and cannot generate representative graph-level features. Recently, there is extensive development of graph pooling algorithms based on the clusters of the embeddings [6, 15, 25]. However, the clusters constructed based on similarity are inconsistent across graphs. This leads to a large degree of freedom in parameters which easily causes overfitting. A semantic-consistent pooling method is therefore needed.

Explaining GNNs. Despite the success of graph neural networks, their poor interpretability of the parameters makes them notoriously recognized as “blackboxes”. With the advances in network attribution methods [29], extensive attempts have been made to open such “blackboxes” [24, 39]. Generating network explanation is an important qualitative step in the WSI analysis since it can highlight the abnormal regions for further investigation. Conventional explainers try to find the associations between the parameters in deep neural networks (or the nodes in GNNs) and the predictions. GNNExplainer [39] is the SOTA method explaining the contributions of node features to the GNN predictions. It trains feature masks on each node and edge feature to minimize the prediction loss of a trained GNN. PGExplainer [24] shares the same objective as GNNExplainer and trains a generative model to generate explanations. Recently, there has been emerging attention in generating causal explanations for GNNs [23, 29], and most of the methods focus on the Granger causality as the explanation

objective. Gem [23] trains explanation generators from the causal perspective. Causal explainers attempt to provide explanations of features that are causal rather than associated with the neural network prediction.

3. Preliminaries

Heterogeneous Graph: A heterogeneous graph is defined by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}, \mathcal{E}, \mathcal{A}$ represent the set of entities (vertices or nodes), relations (edges), and entity types, respectively. And \mathcal{R} represents the space of edge attributes. For $v \in \mathcal{V}$, v is mapped to an entity type by a function $\tau(v) \in \mathcal{A}$. An edge $e = (s, r, t) \in \mathcal{E}$ links the source node s and the target node t , and r is mapped to an edge attribute by a function $\phi(e) = r \in \mathcal{R}$. Every node v has a d -dimensional node feature $x \in \mathcal{X}$, where \mathcal{X} is the embedding space of node features.

Granger Causality [13, 23]: Let \mathcal{I} be all the available information and \mathcal{I}_{-X} be the information excluding variable X . If we can make a better prediction of Y using \mathcal{I} than using \mathcal{I}_{-X} , we conclude that X Granger-causes Y .

WSI Classification: Given a WSI X and a heterogeneous graph \mathcal{G} constructed from X , we wish to predict the label y with a GNN model \mathcal{M} . We also aim to assign an importance score $f(v)$ to each node $v \in \mathcal{V}$ in \mathcal{G} as the causal contribution of each patch to the prediction for localization.

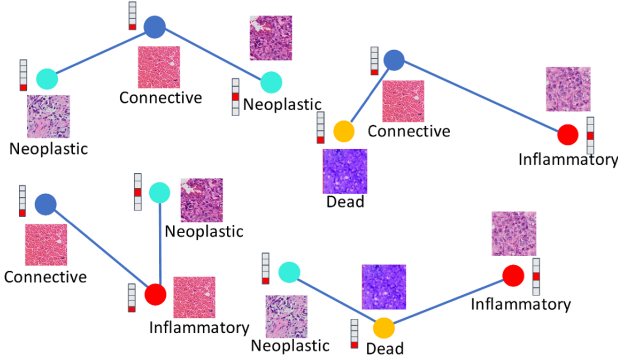


Figure 3. Examples of introduced meta-relations in a heterogeneous graph constructed from a WSI.

4. Methodology

4.1. Heterogeneous Graph Construction

We introduce our methodology of modeling the WSI with a heterogeneous graph. Figure 2 presents the overall workflow of our proposed framework. We adopt the commonly used OTSU thresholding algorithm [5] and sliding window strategy to crop each WSI into non-overlapping patches. Uninformative patches with backgrounds are removed. These patches define the nodes of the graph constructed. To define the corresponding node type, we use HoverNet [12] pretrained on the PanNuke dataset [8] to classify the patches into predefined types. HoverNet detects nuclei in each patch and assigns types to these nuclei. By majority votes, we take the most frequently predicted nucleus type to be the type of the patch. Figure 1 presents an example of a WSI with patches selected from the OTSU and node types generated by HoverNet [12]. We use a pre-trained feature encoder (i.e., KimiaNet [26]) to obtain the embeddings of each patch, which serves as the features of each node in the heterogeneous graph.

Based on the nodes and node features, we define the edges and edge attributes between the patches. For each node $v \in \mathcal{V}$, we use the k -nearest neighbor algorithm to find k nodes that have the most similar features to that node, and connect edges between node v and these neighboring nodes. For each edge, we compute the Pearson R correlation between the head and tail node features as the edge attributes. The edge attributes introduce heterogeneity in edges and highlight meta-relations in the WSI. We adopt data augmentations (e.g., randomly removing some edges) during training to alleviate the potential noises introduced by the edge attributes. As a result, we obtain a heterogeneous graph \mathcal{G} with heterogeneity introduced by different node types and edge attributes. As shown in Figure 3, a heterogeneous graph outlines the meta-relations between the nuclei in a WSI. Mining these meta-relations can reveal the structural interactions between the cells, leading to improved performances on different tasks.

4.2. Heterogeneous Edge Attribute Transformer

The conventional graph attention mechanism is incapable of tackling the heterogeneity of the graph. Inspired by the transformer architecture [31] and its extension on graphs [16, 17, 40], we propose a new graph aggregation layer, named the Heterogeneous Edge Attribute Transformer (HEAT) layer, to aggregate the structural relations between biological entities in the built heterogeneous graph. We explicitly incorporate the node types and continuous edge features into the aggregation process, which guides the learning of edge similarities. Our proposed architecture also generalizes the existing architecture to incorporate continuous or high-dimensional edge attributes and simplifies the use of linear layers to avoid overfitting led by model over-parameterizations.

For each edge $e = (s, r, t)$ and each attention head i , we project the target node t into a query vector with a linear projection layer $\mathbf{W}_{\tau(s)}^i$, and the source node into a key vector with $\mathbf{W}_{\tau(t)}^i$. We also compute the value vector $\mathbf{h}_{\text{value}}^i$ of each source node by the same projection layer $\mathbf{W}_{\tau(s)}^i$

$$\begin{aligned} \mathbf{h}_{\text{key}}^i &= \mathbf{W}_{\tau(s)}^i H_s^{(l-1)}, & \mathbf{h}_{\text{query}}^i &= \mathbf{W}_{\tau(t)}^i H_t^{(l-1)}, \\ \mathbf{h}_{\text{value}}^i &= \mathbf{W}_{\tau(s)}^i H_s^{(l-1)}, \end{aligned}$$

where $H_v^{(l-1)}$ is the input node feature for node $v \in \mathcal{V}$ from the $(l-1)$ -th layer. These projection layers can project node features of various node types into a node-type-invariant embedding space. The edge attributes from the $(l-1)$ -th layer $h_e^{(l-1)}$ are also projected to $h'_e = W_{\text{edge}} h_e^{(l-1)}$ by a linear projection layer W_{edge} . After projecting the node embeddings, we compute the dot-product similarity between the query and key vectors and further multiply the linear transformed edge attribute to the similarity score to incorporate the edge attributes in \mathcal{G} . We then concatenate the scores from each head and take the softmax of the score (i.e., over-weights of incoming edges for all neighboring nodes) to obtain the final attention scores to the value vector $\mathbf{h}_{\text{value}}^i$,

$$\begin{aligned} \text{Attention}(e) &= \text{softmax}_{\forall s \in N(t)} \left(\left\| \text{ATT}(e, i) \right\|_{i \in [1, h]} \right), \\ \text{ATT}(e, i) &= \left(\mathbf{h}_{\text{key}}^i h'_e \mathbf{h}_{\text{query}}^i \right) / \sqrt{d}, \end{aligned}$$

where $N(t)$ is the set of all the source nodes pointing to target node t , d is the dimension of node embeddings, $\text{ATT}(e, i)$ represents the i -head attention score of edge e , $\|_{i \in [1, h]}$ is the concatenation operator concatenating the attention scores from all heads and $\text{Attention}(e)$ represents the final attention score of the edges aggregating all the heads. We multiply the attention score obtained by the value vector to obtain the output features. By doing so, the output features contain both the node-type and edge-attribute-specific information. Hence the HEAT layer can capture the

Algorithm 1 The HEAT algorithm.

Input:

Heterogeneous graph \mathcal{G}_{l-1} with node features $\{H_i^{(l-1)}, \forall i \in \mathcal{V}\}$ and edge attribute $\{h_e^{(l-1)}, \forall e \in \mathcal{E}\}$;
Node-type specific projection layers $\{W_a^i, \forall a \in \mathcal{A}\}$
Edge attribute transformation layer W_{edge} .

Output: The updated graph \mathcal{G}_l with node features $\{H_i^{(l)}, \forall i \in \mathcal{V}\}$, and the edge features $\{h_e^{(l)}, \forall e \in \mathcal{E}\}$

- 1: Initialize projection layers for each node type
 - 2: **for** $e = (s, t) \in \mathcal{E}$ **do**
 - 3: $\mathbf{h}_{\text{key}}^i = W_{\tau(s)}^i H_s^{(l-1)}$ ▷ Project the source node
 - 4: $\mathbf{h}_{\text{value}}^i = W_{\tau(s)}^i H_s^{(l-1)}$ ▷ Compute value vector
 - 5: $\mathbf{h}_{\text{query}}^i = W_{\tau(t)}^i H_t^{(l-1)}$ ▷ Project the target node
 - 6: $h'_e \leftarrow W_{\text{edge}} \cdot h_e^{(l-1)}$ ▷ Project the edge attribute
 - 7: $\text{ATT}(e, i) = \left(\mathbf{h}_{\text{key}}^i h'_e \mathbf{h}_{\text{query}}^i \right) / \sqrt{d}$
 - 8: $\text{Attention}(e) = \text{softmax}(\|_{i \in [1, h]} \text{ATT}(e, i))_{\forall s \in N(t)}$
 - 9: $h_e^{(l)} \leftarrow h'_e$ ▷ Compute latent edge features
 - 10: **end for**
 - 11: **for** $t \in \mathcal{V}$ **do**
 - 12: $H_t^{(l)} = \bigoplus_{\forall s \in N(t)} (\|_{i \in [1, h]} \mathbf{h}_{\text{value}}^i \cdot \text{Attention}(e))$
 - 13: **end for**
 - 14: **return** \mathcal{G}_l
-

structural information in \mathcal{G} by transforming the node features from different node types. It can also model different semantic relations since edge attributes are included in the aggregation.

Finally, we perform target-specific aggregation to update the feature of each target node by averaging its neighboring node features. We concatenate all h attention heads to obtain the attention vector for each pair of source and target nodes. For each target node t , we conduct a softmax operation on all the attention vectors from its neighboring nodes and then aggregate the information of all neighboring source nodes of t together. The updated node features $H_t^{(l)}$ for \mathcal{G}_l can be represented as

$$H_t^{(l)} = \bigoplus_{\forall s \in N(t)} \left(\big\|_{i \in [1, h]} \mathbf{h}_{\text{value}}^i \cdot \text{Attention}(e) \right),$$

where \bigoplus is an aggregation operator (e.g., mean aggregation). The updated graph \mathcal{G}_l is returned as the output of the l -th HEAT layer. Algorithm 1 demonstrates the overall process of our proposed HEAT layer.

4.3. Pseudo-label Graph Pooling

We introduce a novel pooling method — pseudo-label (PL) pooling, to aggregate information with respect to the pseudo-labels (i.e., node types) predicted from a pretrained teacher network (e.g., HoverNet [12]). Unlike conventional

Algorithm 2 The PL-Pool Algorithm

Input: Heterogeneous graph \mathcal{G} with node features $\{H_i, \forall i \in \mathcal{V}\}$ and node type set \mathcal{A} .

Output: The pooled graph-level feature $\mathbf{S} \in \mathbb{R}^{|\mathcal{A}| \times d}$.

- 1: Initialize readout layers for each node type $a \in \mathcal{A}$.
 - 2: Initialize aggregate feature matrix \mathbf{S} .
 - 3: **for** $a \in \mathcal{A}$ **do**
 - 4: $X_a \leftarrow$ feature matrix of nodes of type a
 - 5: $\mathbf{h}_a \leftarrow \text{readout}_a(X_a)$ ▷ Pool feature with readout layer
 - 6: $\mathbf{S}_a = \mathbf{h}_a$ ▷ Assign pooled feature to the a -th row of \mathbf{S}
 - 7: **end for**
 - 8: **return** \mathbf{S}
-

methods of pooling features based on clusters, we define clusters using a pretrained node classifier. Pooling from pseudo-labels ensures the semantic consistency in cluster definitions and distills the context knowledge (e.g., nuclei features) from the teacher network. Specifically, for each node type a , we pool all node features belonging to type a into a single vector \mathbf{h}_a with a readout layer. The pooled features from each node type are then aggregated into a feature matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{A}| \times d}$. The graph level feature is then determined by another readout layer (e.g., mean readout).

Algorithm 2 presents the workflow of the proposed PL Pooling. By pooling with the pseudo-labels, we are able to cluster patch representation according to nuclei types, such that the graph-level features are enhanced with the prior knowledge on nuclei type distributions. The detailed mechanism of the PL Pool is presented in the supplementary materials. We also perform an ablation study in Table 4 and show that PL Pooling outperforms existing pooling methods in cancer classification tasks.

4.4. Prior Knowledge Regularization

Here we discuss the motivation for introducing prior knowledge in our proposed HEAT and PL pooling algorithms. In the context of WSI analysis when the data are scarce, while data distributions are sparse and high-dimensional. The curse of high dimensionality makes the sampling distributions difficult to approximate the properties of true distributions of the WSIs. This leads to a significant gap between training and testing distributions. Hence regularization techniques are needed to reduce the model variance and mitigate performance deterioration when transferring the model from training to testing environments. Since WSI data contain enriched prior knowledge (e.g., the interaction among different cell types), integrating such knowledge into the framework regularizes the model, such that the testing performance improves. Therefore, we design the above two designs by integrating prior

		Cancer Staging (Four Stages)			Cancer Classification		
	Model	AUC	Accuracy	Macro-F1	AUC	Accuracy	Macro-F1
TCGA-COAD	ABMIL [18]	53.8 (3.7)	19.2 (7.8)	35.8 (4.4)	97.7 (2.3)	98.3 (0.9)	95.8 (2.2)
	DSMIL [21]	59.3 (1.4)	35.7 (5.7)	37.9 (2.8)	99.7 (0.2)	98.6 (0.5)	96.9 (0.9)
	ReMix [38]	58.3 (1.5)	33.9 (7.8)	24.8 (7.5)	94.3 (3.4)	96.0 (4.6)	92.8 (5.9)
	PatchGCN [5]	62.5 (4.9)	38.2 (3.1)	38.5 (5.7)	91.1 (5.3)	97.1 (2.0)	98.8 (1.0)
	GTNMIL [41]	54.2 (2.6)	29.3 (1.4)	24.3 (3.9)	97.3 (2.6)	98.1 (1.3)	95.9 (2.4)
	H ² -MIL [15]	58.6 (2.7)	38.5 (5.4)	33.0 (5.0)	99.7 (0.4)	99.2 (0.5)	97.4 (1.7)
	HEAT (Ours)	63.4 (2.5)	40.0 (2.1)	41.3 (2.7)	99.9 (0.2)	99.9 (0.3)	99.2 (0.4)
TCGA-BRCA	ABMIL [18]	54.7 (4.6)	19.0 (10.0)	23.9 (3.2)	97.3 (1.7)	98.3 (1.1)	97.3 (1.6)
	DSMIL [21]	51.4 (4.7)	18.3 (14.9)	23.2 (2.3)	98.7 (0.5)	95.6 (1.4)	93.3 (2.0)
	ReMix [38]	58.8 (2.2)	35.6 (16.2)	27.6 (5.8)	96.1 (0.7)	95.8 (2.6)	93.0 (3.4)
	PatchGCN [5]	50.3 (0.2)	41.6 (0.5)	25.1 (0.3)	96.2 (1.7)	98.2 (0.8)	98.4 (0.8)
	GTNMIL [41]	53.0 (3.7)	41.3 (4.4)	25.1 (2.3)	94.7 (1.0)	94.5 (0.2)	93.7 (1.7)
	H ² -MIL [15]	52.1 (7.2)	53.7 (2.6)	21.2 (2.5)	97.9 (2.7)	98.0 (1.5)	97.6 (2.2)
	HEAT (ours)	61.9 (3.8)	55.8 (6.4)	27.7 (16.3)	98.8 (0.7)	98.3 (0.5)	99.5 (0.7)

Table 1. Cancer staging and classification results [%] of various methods on TCGA-COAD and TCGA-BRCA datasets.

knowledge into the feature aggregation procedure. Specifically, for the HEAT layer, we integrate the prior knowledge of node type and node attributes when extracting node-level features. For PL Pooling, we pool node-level features using prior definitions on node clusters. Moreover, we perform data augmentations (e.g., random pruning on edges and nodes) to regularize the learning from training distributions. Besides that, other regularization such as imposing a Gaussian prior on the model weights (i.e., using a Bayesian neural network) would also achieve the goal.

4.5. Causal-driven Localization

We make use of the Granger causality to outline causal regions in the WSI with the causal graph explainer [23]. Given a trained GNN model \mathcal{M} , the causal contribution of each node v is given by

$$\Delta_{\delta,v} = \mathcal{L}(y, \tilde{y}_{\mathcal{G}}) - \mathcal{L}(y, \tilde{y}_{\mathcal{G}\setminus\{v\}}), \quad (1)$$

where y is the true label and $\tilde{y}_{\mathcal{G}} = \mathcal{M}(\mathcal{G})$ and $\tilde{y}_{\mathcal{G}\setminus\{v\}} = \mathcal{M}(\mathcal{G}\setminus\{v\})$ are the predicted labels from \mathcal{M} with input graphs \mathcal{G} and $\mathcal{G}\setminus\{v\}$, respectively. The causality heatmap of the patches can then be visualized with the causal contribution computed for each patch (i.e., node). Addressing causality in instance interpretation can adjust for observational and selection biases, which would improve the explanation accuracy. Moreover, the causal property of the explainer could facilitate pathologists to find out potential biomarkers for diagnosis and prognosis by highlighting the patches with clinical relevance in the WSI.

5. Experiments

5.1. Datasets

We use WSIs from the public TCGA-COAD (cancer staging task: 1304 cases, classification task: 1434 cases),

Model	AUC	Accuracy	Macro-F1
ABMIL [18]	79.5 (7.5)	80.3 (8.4)	81.3 (7.4)
DSMIL [21]	92.5 (1.7)	87.3 (2.0)	86.3 (2.0)
ReMix [38]	92.5 (7.2)	90.0 (8.1)	90.3 (7.7)
PatchGCN [5]	88.6 (3.5)	92.1 (2.3)	92.3 (2.4)
GTNMIL [40]	89.7 (4.7)	81.2 (4.8)	89.2 (4.9)
H ² -MIL [15]	92.1 (3.9)	88.2 (5.8)	88.0 (5.8)
HEAT (ours)	92.8 (2.5)	92.7 (2.2)	93.3 (1.9)

Table 2. Cancer typing results [%] of our method compared to various methods on the TCGA-ESCA dataset.

TCGA-BRCA (cancer staging task: 1328 cases, classification task: 1712 cases), and TCGA-ESCA (typing task: 213 cases) from the TCGA project [35] and Camelyon 16 [3] as the benchmark datasets. On average, around 300 patches are sampled from each WSI in the TCGA datasets (around 5,000 for Camelyon 16), where each patch corresponds to a node in the final heterogeneous graph. For the TCGA-COAD and the TCGA-BRCA datasets, we conduct two tasks for the benchmark methods — cancer staging and cancer classification. For the cancer staging task, all the cases are divided into the “Stage I”, “Stage II”, “Stage III”, and “Stage IV” classes. For the cancer classification task, all the cases are divided into the “Normal” and “Tumor” classes. For the cancer typing task, we use TCGA-ESCA dataset where all the cases are divided into two classes i.e., “Type I: adenocarcinoma” and “Type II: squamous cell carcinoma”. We also evaluate the localization ability of our framework on the Camelyon 16 dataset, as this dataset provides the tumor mask annotations. A detailed summary of datasets is provided in supplementary materials.

5.2. Implementation Details

The proposed framework is implemented in Python with the *Pytorch* library on a server equipped with four NVIDIA

GNN Architecture	AUC	Accuracy	Macro-F1
GCN [36]	90.8	90.9	90.0
GAT [32]	85.8	86.4	88.9
GIN [37]	91.6	90.9	83.3
HetRGCN [28]	82.5	83.3	88.9
HGT [16]	87.8	87.5	83.3
HEAT (ours)	92.8	92.7	93.2

Table 3. Cancer typing results [%] of our method compared to various GNN architectures on the TCGA-ESCA dataset.

Pooling Method	AUC	Accuracy	Macro-F1
Sum pooling	95.5	99.3	99.2
Max pooling	95.1	98.6	99.2
Mean pooling	97.7	95.8	99.8
Global attention pooling [22]	94.7	97.9	99.2
IH-Pool [15]	99.3	97.2	88.1
ASAP [25]	99.2	98.6	95.1
PL-Pool (ours)	99.6	99.3	99.8

Table 4. Cancer classification results [%] on TCGA-COAD of our pooling method to various comparable pooling methods using GCN and KimiaNet feature encoder.

TESLA V100 GPUs. We use *openslide* [11] as the tool to process the WSIs. The dropout ratio of each dropout layer is selected as 0.2. All models are trained with 150 epochs with early stopping. The batch size is selected as 2. We adopt the cross-entropy loss to train the network for classification tasks. We use the Adam optimizer to optimize the model with a learning rate of 5×10^{-5} and a weight decay of 1×10^{-5} . We perform data augmentations on the training graphs by randomly dropping the edges and nodes, and adding Gaussian noises to the node and edge features.

5.3. Experiment Settings and Evaluation Metrics

We compare our method with an array of SOTA methods, including MIL or graph-based methods. We use five-fold cross-validation to evaluate the overall performance of our framework and other methods. We used the pretrained KimiaNet as the feature extraction for all methods for a fair comparison. The details of compared methods are listed below.

- ABMIL [18]: a MIL framework aggregating bag-level instance information by the attention mechanism.
- DSMIL [21]: a dual-stream multiple instance learning method using max pooling and attention to aggregate the signals from the individual patches.
- ReMix [38]: a general and efficient MIL’s based framework for WSI analysis that takes the advantage of data augmentation and reduces method to produce rich features.
- PatchGCN [5]: a hierarchical graph-based model on survival data with patient-level and WSI-level aggregations. We adapt this method as a GCN model with global attention pooling [22].

Balanced dataset	AUC	Accuracy	Macro-F1
TCGA-COAD	99.1 (1.8)	99.1 (1.8)	99.2 (1.7)
TGCA-BRCA	98.7 (2.5)	98.7 (2.5)	98.7 (2.6)

Table 5. Cancer classification results [%] of our method on TCGA-COAD and TCGA-BRCA balanced datasets.

- GTNMIL [41]: a graph-based MIL method based on the graph transformer network [40].
- H²-MIL [15]: a tree-graph-based multiple instance learning method that utilizes different magnification levels to represent hierarchical features.

For the cancer staging, classification and typing tasks, we use AUC, classification accuracy, and macro F-1 score as the evaluation metrics. Percentage [%] values are reported for each of the metrics. Standard errors are reported in brackets. For all metrics, a higher value indicates a better performance. Detailed definitions of the evaluation metrics can be found in the supplementary materials.

5.4. Comparison with Other Methods

Quantitative Results. Table 1 shows the cancer staging and classification results on the TCGA-COAD and the TCGA-BRCA datasets, and Table 2 presents cancer typing results on the TCGA-ESCA dataset. Compared to graph-based WSI analysis methods [5, 15, 41], our method demonstrates improved performance, which indicates our graph modeling method potentially better represents the interaction of patches in a WSI than existing graph-based methods. We also observe that aggregation on a graph of instances is more effective than aggregation on bags of instances in the staging tasks, which implies graph-based methods are more capable of capturing the global information of WSI for staging tasks than conventional MIL methods [18, 21, 38]. We further compare HEAT on the BRCA subtyping task with a recent SOTA method on WSI — hierarchical image pyramid transformer (HIPT) [4]. Our method achieves an AUC of 89.69 (SD: 3.63), which outperforms the AUC of 87.4 (SD: 6.0) by HIPT.

Additionally, we perform a t-test on the AUCs to demonstrate the statistical significance of our improvements over the SOTA methods, for which the results are presented in Table 6. We observe that the improvements are statistically significant over most of the baseline methods under the 0.05 significance level.

Methods	COAD-S	COAD-C	BRCA-S	BRCA-C
ABMIL	1.91e-16	0	5.75e-5	6.08e-6
DSMIL	0.0005	0.0333	3.26e-9	0.3790
ReMix	1.36e-5	0	0.0759	1.2e-16
PatchGCN	0.2899	0	5.14e-11	2.74e-15
GTNMIL	2.7e-15	0	5.58e-7	2.94e-35
H ² -MIL	4.5e-5	0.0343	3.94e-8	0.0082

Table 6. P-values of two-sample t-tests on AUCs between our method and baselines (S: cancer staging; C: cancer classification).

Qualitative Results. We compute the causal contribution of each patch using Equ. (1). We visualize the patch image associated with that node to outline the causal regions related to the predictions. We also compare our causal explanation method to numerous baseline graph interpretation methods based on associations [39]. Figures in the supplementary materials present the explanation results with different graph explainers on the Camelyon 16 dataset. It is observed that using an association-based explainer provides a smooth heatmap where many regions are highlighted as important. A such heatmap is less accurate in localizing the tumor regions and pathologists still need to traverse a large number of abnormal regions suggested by the explainer to identify tumor regions. On the contrary, we observe that using a causal explainer can outline the tumor regions in the WSIs more accurately, with the heatmap more concentrated on the ground-truth tumor regions compared to association-based explainers (e.g., GNNExplainer [39]).

5.5. Analysis of Our Framework

Effectiveness of Heterogeneous Graph Construction. We compare our method with other SOTA GNNs [16, 28, 32, 36, 37] to evaluate the effectiveness of our heterogeneous graph construction. For heterogeneous graph transformer (HGT) [16] and HetRGCN [28], we define the discrete edge types — each relation either has the “positive” type representing positive correlations between the nodes of the edge, or the “negative” type representing negative correlations. Table 3 presents cancer typing results of our method compared to various SOTA GNN aggregation methods on the TCGA–ESCA dataset. Not only our method outperforms SOTA homogeneous GNN architectures [32, 36, 37], but it is also superior to some recently heterogeneous GNN architectures [16, 28]. This implies the advantage of our proposed architecture for graph-based WSI analysis.

Analysis of Different Pooling Strategies. We compare our proposed pooling strategy to a variety of comparable pooling methods, including basic pooling methods, such as sum/max/mean poolings and advanced pooling strategies [15, 25]. Table 4 presents the comparison results of cancer classification on TCGA–COAD dataset. We fix the model architecture to be GCN [36] and the feature encoder as KimiaNet [26]. It is observed that our pooling strategy outperforms the competitors, which validates the advantage of using semantic-consistently defined clusters in pooling.

Performance on Different Class Distributions. We observe the WSI datasets for cancer classification is imbalanced (i.e., approximately ten cancer WSIs to one normal WSI). We thus compose a balanced dataset (i.e., normal:cancer = 1:1) with the undersampling strategy to study how the difference in class distributions affect the performance of our model. Table 5 presents the comparison. It

is observed that our method achieves similar performance with the unbalanced setting (See Table 1).

Generalizability. The pretrained features are a key component of our proposed framework. As the pretrained embedding models are from a diverse WSI context, they can extract good features from most of the WSI datasets. Because the PanNuke dataset [9] (used to pretrain the HoverNet node type classifier) contains WSIs of most of the common cancer types, this leads to a broad generalization of HoverNet. Furthermore, one may adopt contrastive learning to fine-tune the pretrained models to improve their generalizability to new datasets in potential deployment scenarios.

Accuracy of HoverNet. The performance of the HoverNet classifier would influence the sensitivity of our framework. Since the PanNuke dataset contains WSIs of most of the common cancer types and cohorts of the TCGA dataset (e.g., COAD), there are domain overlaps between them. Hence the HoverNet trained on the PanNuke dataset can be transferred to the TCGA dataset for patch types classification with good performance. Furthermore, we perform cancer classification on COAD using node types generated by unsupervised K-means clustering. The performance (AUC: 98.5) is lower than that using HoverNet predicted node types (AUC: 99.9). This demonstrates that incorporating the pretrained HoverNet outperforms unsupervised methods and improves WSI analysis.

6. Conclusion

We present a novel heterogeneous graph-based framework for WSI analysis. By modeling WSI as a heterogeneous graph with various node types and edge attributes, our method not only leverages the locality information, but also mines the complex relational information of WSI. We further design a novel heterogeneous edge attribute transformer architecture to aggregate the structural information in the graph and a semantic consistent pooling method to address the potential over-parameterization problems in conventional pooling. We provide a causal explanation mechanism to highlight the causal contributions of the instances to improve the clinical usability of our work. Extensive experiments on public datasets validate the effectiveness of our proposed framework and our framework could be adapted to other graph-based computer vision tasks, such as 3D point cloud analysis and anomaly detection.

Acknowledgement. We thank the anonymous reviewers and the area chair for their insightful comments on our manuscript. This work was partially supported by the Research Grants Council of Hong Kong (17308321), the Theme-based Research Scheme (T45-401/22-N), the National Natural Science Fund (62201483), and the HKU-TCL Joint Research Center for Artificial Intelligence sponsored by TCL Corporate Research (Hong Kong).

References

- [1] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. A survey on graph-based deep learning for computational histopathology. *Computerized Medical Imaging and Graphics*, page 102027, 2021. [2](#)
- [2] Yogesh Balaji, Mohammadmahdi Sajedi, Neha Mukund Kalibhat, Mucong Ding, Dominik Stöger, Mahdi Soltanolkotabi, and Soheil Feizi. Understanding over-parameterization in generative adversarial networks. In *International Conference on Learning Representations*, 2020. [2](#)
- [3] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermesen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. [2](#), [6](#)
- [4] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022. [7](#)
- [5] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 339–349. Springer, 2021. [2](#), [4](#), [6](#), [7](#)
- [6] Moshe Eliasof and Eran Treister. Diffgcn: Graph convolutional networks via differential operators and algebraic multigrid pooling. *Advances in neural information processing systems*, 33:18016–18027, 2020. [3](#)
- [7] Alton B Farris, Juan Vizcarra, Mohamed Amgad, Lee AD Cooper, David Gutman, and Julien Hogan. Artificial intelligence and algorithmic computational pathology: an introduction with renal allograft examples. *Histopathology*, 78(6):791–804, 2021. [2](#)
- [8] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. Pannuke: an open pancreatic histology dataset for nuclei instance segmentation and classification. In *European congress on digital pathology*, pages 11–19. Springer, 2019. [4](#)
- [9] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benes, Simon Graham, Mostafa Jahanifar, Syed Ali Khuram, Ayesha Azam, Katherine Hewitt, and Nasir Rajpoot. Pannuke dataset extension, insights and baselines. *arXiv preprint arXiv:2003.10778*, 2020. [8](#)
- [10] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9211–9219, 2019. [2](#)
- [11] Adam Goode, Benjamin Gilbert, Jan Harkes, Drazen Jukic, and Mahadev Satyanarayanan. Openslide: A vendor-neutral software foundation for digital pathology. *Journal of pathology informatics*, 4, 2013. [7](#)
- [12] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019. [2](#), [4](#), [5](#)
- [13] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969. [2](#), [3](#)
- [14] Yuqi Gu and David B Dunson. Bayesian pyramids: Identifiable multilayer discrete latent structure models for discrete data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 2021. [2](#)
- [15] Wentai Hou, Lequan Yu, Chengxuan Lin, Helong Huang, Rongshan Yu, Jing Qin, and Liansheng Wang. H2-mil: Exploring hierarchical representation with heterogeneous multiple instance learning for whole slide image analysis. 2022. [2](#), [3](#), [6](#), [7](#), [8](#)
- [16] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710, 2020. [2](#), [4](#), [7](#), [8](#)
- [17] Tiancheng Huang, Ke Xu, and Donglin Wang. Da-hgt: Domain adaptive heterogeneous graph transformer. *arXiv preprint arXiv:2012.05688*, 2020. [4](#)
- [18] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. [6](#), [7](#)
- [19] Ryosuke Kojima, Shoichi Ishida, Masateru Ohta, Hiroaki Iwata, Teruki Honma, and Yasushi Okuno. kgcn: a graph-based deep learning framework for chemical structures. *Journal of Cheminformatics*, 12:1–10, 2020. [2](#)
- [20] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *Proceedings of the 2nd SysML Conference*, 2019. [2](#)
- [21] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021. [6](#), [7](#)
- [22] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *Proceedings of ICLR'16*, 2015. [7](#)
- [23] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021. [3](#), [6](#)
- [24] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020. [3](#)
- [25] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical

- graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5470–5477, 2020. [3](#), [7](#), [8](#)
- [26] Abtin Riasatian. Kimianet: Training a deep network for histopathology using high-cellularity. Master’s thesis, University of Waterloo, 2020. [2](#), [4](#), [8](#)
- [27] Joel Saltz, Rajarsi Gupta, Le Hou, Tahsin Kurc, Pankaj Singh, Vu Nguyen, Dimitris Samaras, Kenneth R Shroyer, Tianhao Zhao, Rebecca Batiste, et al. Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. *Cell reports*, 23(1):181–193, 2018. [2](#)
- [28] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018. [2](#), [7](#), [8](#)
- [29] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019. [3](#)
- [30] David Tellez, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. Neural image compression for gigapixel histopathology image analysis. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):567–578, 2019. [2](#)
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [4](#)
- [32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2017. [2](#), [7](#), [8](#)
- [33] Shujun Wang, Yaxi Zhu, Lequan Yu, Hao Chen, Huangjing Lin, Xiangbo Wan, Xinjuan Fan, and Pheng-Ann Heng. Rmdl: Recalibrated multi-instance deep learning for whole slide gastric image classification. *Medical image analysis*, 58:101549, 2019. [2](#)
- [34] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019. [2](#)
- [35] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013. [2](#), [6](#)
- [36] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016. [2](#), [7](#), [8](#)
- [37] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018. [2](#), [7](#), [8](#)
- [38] Jiawei Yang, Hanbo Chen, Yu Zhao, Fan Yang, Yao Zhang, Lei He, and Jianhua Yao. Remix: A general and efficient framework for multiple instance learning based whole slide image classification. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*, 2022. [6](#), [7](#)
- [39] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019. [3](#), [8](#)
- [40] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019. [2](#), [4](#), [6](#), [7](#)
- [41] Yi Zheng, Rushin H Gindra, Emily J Green, Eric J Burks, Margrit Betke, Jennifer E Beane, and Vijaya B Kolachalama. A graph-transformer for whole slide image classification. *IEEE transactions on medical imaging*, 41(11):3003–3015, 2022. [2](#), [6](#), [7](#)
- [42] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018. [2](#)