MulGT: Multi-Task Graph-Transformer with Task-Aware Knowledge Injection and Domain Knowledge-Driven Pooling for Whole Slide Image Analysis

Weiqin Zhao[1], **Shujun Wang**[2], **Maximus Yeung**[1], **Tianye Niu**[3], **Lequan Yu**[1,*]

[1] The University of Hong Kong, Hong Kong SAR, China
[2] University of Cambridge, Cambridge, UK
[3] Shenzhen Bay Laboratory, Shenzhen, China
wqzhao98@connect.hku.hk, sw991@cam.ac.uk, mcfyeung@pathology.hku.hk, niuty@szbl.ac.cn, lqyu@hku.hk

Abstract

Whole slide image (WSI) has been widely used to assist automated diagnosis under the deep learning fields. However, most previous works only discuss the SINGLE task setting which is not aligned with real clinical setting, where pathologists often conduct multiple diagnosis tasks simultaneously. Also, it is commonly recognized that the multi-task learning paradigm can improve learning efficiency by exploiting commonalities and differences across multiple tasks. To this end, we present a novel multi-task framework (*i.e.*, MulGT) for WSI analysis by the specially designed Graph-Transformer equipped with Task-aware Knowledge Injection and Domain Knowledge-driven Graph Pooling modules. Basically, with the Graph Neural Network and Transformer as the building commons, our framework is able to learn task-agnostic low-level local information as well as task-specific high-level global representation. Considering that different tasks in WSI analysis depend on different features and properties, we also design a novel Task-aware Knowledge Injection module to transfer the task-shared graph embedding into task-specific feature spaces to learn more accurate representation for different tasks. Further, we elaborately design a novel Domain Knowledge-driven Graph Pooling module for each task to improve both the accuracy and robustness of different tasks by leveraging different diagnosis patterns of multiple tasks. We evaluated our method on two public WSI datasets from TCGA projects, *i.e.*, esophageal carcinoma and kidney carcinoma. Experimental results show that our method outperforms single-task counterparts and the state-of-the-art methods on both tumor typing and staging tasks.

Introduction

Histopathology analysis is the gold standard method for cancer diagnosis and prognosis. Experienced pathologists can provide accurate analysis of biopsy specimens based on whole slide image (WSI), *i.e.*, the high-resolution digitalization of the entire histology slide (Khened et al. 2021; Pataki et al. 2022). However, analyzing the WSIs is time-consuming and laborious due to the massive size of the WSIs and the complex colors and patterns of different tissue structures. To elevate the precision and speed of the examination, extensive research tools have been developed for automate computational WSI inspection (Wang et al. 2019b,a; Coudray et al. 2018).

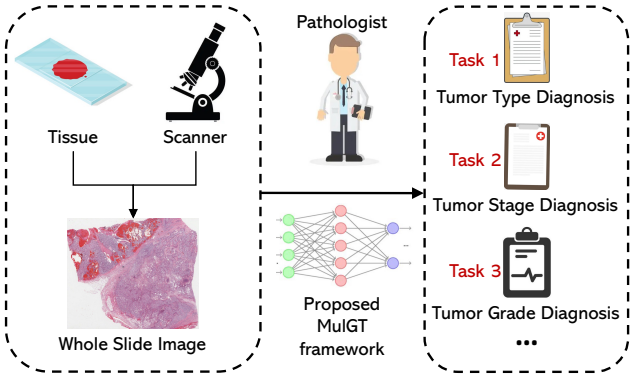

Figure 1: Illustration of the multi-task learning setting for WSI analysis.

Due to the powerful expressivity of neural networks, many deep learning-based methods have been proposed for WSI analysis recently. However, WSIs usually have a huge size (*e.g.*, $150,000 \times 150,000$), and it will be expensive to obtain detailed pixel-level annotations. To overcome such challenges, multiple instance learning (MIL) (Maron and Lozano-Pérez 1998) becomes a promising direction to analyze WSI from slide-level annotations. Specifically, MIL-based approaches first extract the feature embeddings of image tiles (*i.e.* patches) with a Convolution Network (CNN) (He et al. 2016; Riasatian et al. 2021) or Vision Transformer Network (ViT) (Chen et al. 2022). Then, the feature embeddings are fed into an aggregation network to produce the slide-level predictions. Various network architectures have been employed to aggregate the information, including Graph Neural Network (GNN) (Hou et al. 2022; Guan et al. 2022), Transformer network (Chen et al. 2022; Wang et al. 2022), and *etc.* Currently, Graph-Transformer architecture has also been introduced into WSI analysis (Zheng et al. 2021) due to its nature to extract both low-level local features and high-level global information from the graphs (*i.e.*WSIs).

However, most of the above works were limited to the setting with a single task, while pathologists often conducts more than one diagnosis results for one particular WSI

*Corresponding author.

(per patient), as shown in Figure 1. Besides, it is believed that multi-task learning paradigm can improve learning efficiency and prediction accuracy by exploiting commonalities and differences across tasks. Although there are also some works (Yang et al. 2020; Vuong et al. 2020; Murthy et al. 2017) discussing multi-task learning in WSI analysis. They were designed for the patch-level prediction tasks instead of the slide-level ones. Therefore, they require patch-level annotations for training and hardly to be extended to the weakly-supervised slide-level label prediction directly. To address the above issues, we present a multi-task Graph-Transformer framework (*i.e.*, MulGT) to conduct multiple slide-level diagnosis tasks simultaneously.

Our framework leverages the architecture of Graph-Transformer from two aspects: (1) learning task-agnostic low-level representation with a shared GNN, and (2) learning task-specific high-level representation with independent Transformer branches. Meanwhile, considering that different tasks in WSI analysis usually require different features and properties of the tissue, we thereby design a novel Task-aware Knowledge Injection module in our framework to transfer the task-shared graph embedding into task-specific feature spaces via the cross-attention mechanism with a set of trainable task-specific latent tokens. Furthermore, to reduce the computation cost, a graph pooling layer is usually adopted between the GNN part and the Transformer part in the Graph-Transformer architecture. However, no attention has been paid to discussing the relationship among tasks or the graph pooling methods. In this paper, we are the first to argue that, in order to boost the performance of the Graph-Transformer architecture, it is necessary to design a task-aware pooling method to meet the different requirements of different downstream tasks. Especially in multi-task learning settings, the graph pooling methods should vary in different task branches if the nature of the tasks is different. Therefore, we elaborately design a novel Domain Knowledge-driven Graph Pooling module in our framework to improve both the accuracy and robustness of different task branches by leveraging the different diagnosis patterns of multiple WSI analysis tasks.

Our main contributions can be summarized as follows.

- We devise a novel multi-task Graph-Transformer for slide-level WSI analysis. Different from methods, our framework conducts multiple diagnosis tasks simultaneously, thus benefiting from learning both the commonalities and differences of multiple tasks. Extensive experiments with promising results on two public WSI datasets validate the effectiveness of our designed framework.
- To learn task-specific features, we design a novel Task-aware Knowledge Injection module to transfer the task-shared feature into task-specific feature spaces via the cross-attention mechanism with the latent tokens that contain task-specific knowledge.
- To import the prior knowledge from different diagnosis patterns of different tasks, we elaborately design a novel Domain Knowledge-driven Graph Pooling module to represent the information of the whole graph more properly for different tasks, facilitating the prediction process and reducing the computation cost.

Related Work

Multiple Instance Learning for WSI. Multiple instance learning (MIL) methods are widely used for WSI analysis and can be categorized into two paradigms: (1) instance-level methods and (2) embedding-level methods (Amores 2013). Generally, instance-level methods typically focus more on local information, while embedding-level methods emphasize global representation. Several recent works adopted attention mechanisms into MIL for WSI analysis for instance aggregation. Particularly, the attention-based approach is able to identify the contribution of different instances during the global aggregation, like ABMIL (Ilse, Tomczak, and Welling 2018), DeepAttnMIL (Yao et al. 2020), and CLAM (Lu et al. 2021). Recently, Graph-based and Transformer-based methods have also been utilized in computational pathology, as WSI instances could be abstracted as nodes of a graph or tokens of Transformer architecture. For example, H2Graph (Hou et al. 2022) built a heterogeneous graph with different resolutions of WSI to learn a hierarchical representation, while HIPT (Chen et al. 2022) introduced a new ViT architecture to learn from the natural image hierarchical structure inherent in WSIs. However, most of the previous works were limited to the single task setting for slide-level analysis.

Multi-task Learning. Multi-task learning (Caruana 1997) jointly optimizes a set of tasks with hard or soft parameter sharing. It is well known that learning multiple tasks simultaneously can offer several advantages, including improved data efficiency and reduced overfitting through the regularization among multiple tasks (Crawshaw 2020). Some previous literature leveraged the relationship among multiple tasks in an explicit way. For example, ML-GCN (Chen et al. 2019) built a directed graph over different object labels to facilitate multi-label image recognition, where each node is one particular object (*i.e.*task) and edges are object correlations. Meanwhile, some works (Kendall, Gal, and Cipolla 2018; Chen et al. 2018) adopted adaptive weights for different tasks to balance the training process, while Liu *et. al.* (2021) introduced gradient-based methods to mitigate the negative transfer across tasks. Especially, RotoGrad (Javaloy and Valera 2021) used a set of rotation matrices to rotate the task-shared features into different feature spaces before the task-specific branches to avoid the gradient conflict among tasks. Partially inspired that transferring task-shared features into different task-specific feature spaces may benefit model learning, this paper designed a Task-aware Knowledge Injection module to differentiate the features in different task branches.

Combining Graph and Transformer. Recently, the Transformer model has been introduced to deal with graph-structured data. According to the relative position of the GNN and Transformer layers, current works could be divided into three architectures (Min et al. 2022): (1) building Transformer blocks on top of GNN blocks; (2) alternately stacking GNN and Transformer blocks (2021); and

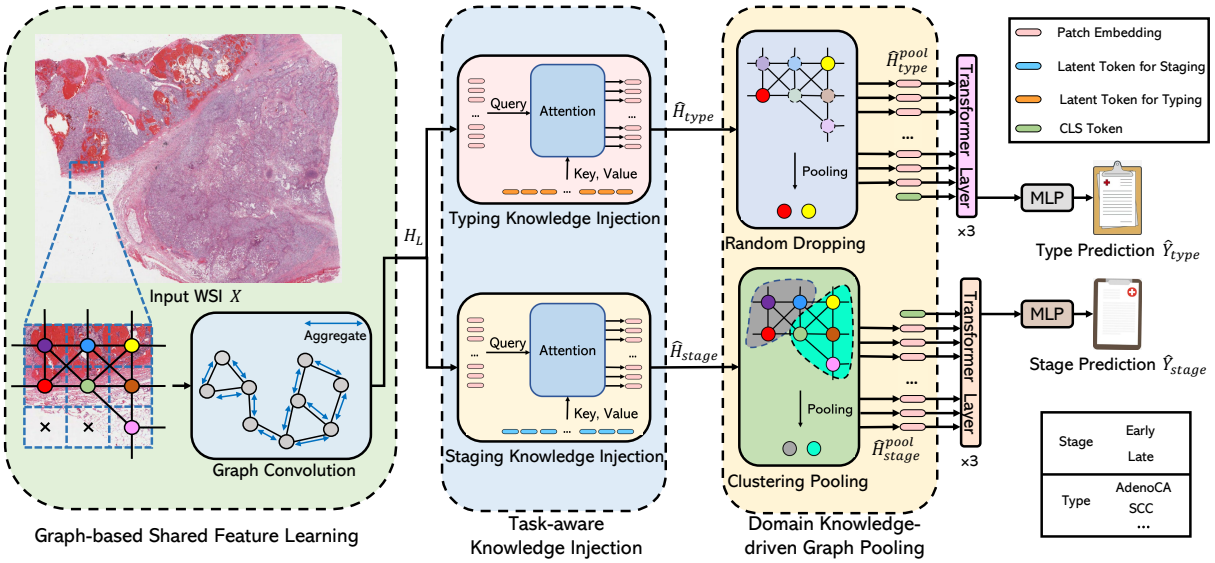

Figure 2: Overview of the proposed MulGT framework. Patches are extracted from WSIs and abstracted as graph nodes. Follow the multi-task learning paradigm, the GNN part served as the task-shared layers to learn task-agnostic low-level local representation, while our proposed Task-aware Knowledge Injection and Domain Knowledge-drive Graph Pooling modules together with the transformer stack served as the task-independent layers to learn accurate high-level global representation.

(3) parallelizing GNN and Transformer blocks (2020). Most works (Rong et al. 2020a; Mialon et al. 2021) adopted the first architecture. Especially, GraphTrans (Wu et al. 2021) applied a permutation-invariant Transformer module after a standard GNN module to learn the high-level and long-range relationships. Graph-Transformer architecture has also been introduced to handle the WSI analysis tasks (Zheng et al. 2021). However, the existing studies are limited to the single task setting and pay no attention to leveraging the domain knowledge from the pathologists for better model design.

Methodology

In this section, we elaborate on our designed multi-task framework for WSI analysis with specially designed Task-aware Knowledge Injection and Domain Knowledge-driven Graph Pooling modules. Figure 2 shows the overview of the proposed framework. Given a WSI X, our framework predicts the labels of two tasks simultaneously: the slide-level tumor typing label $\hat{Y}_{type}$ and staging label $\hat{Y}_{stage}$. Specifically, we first construct graph $\mathcal{G}$ followed by the task-shared Graph Convolution (GC) layer. After that, the framework is divided into two task-specific branches by the corresponding Task-specific Knowledge Injection modules. Further, the transferred task-specific graph representation is fed into the corresponding Domain Knowledge-driven Graph Pooling module for each branch. Finally, a sequence of task-specific Transformer layers followed by MLP are employed to predict slide-level labels of multiple tasks according to the pooled task-specific representations.

Graph-based Shared Feature Learning

As illustrated in Figure 2, given a WSI X under $20\times$ magnification, we first apply the sliding window strategy to crop X into numerous image tiles without overlap. Then we construct a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ represents the extracted feature embeddings of the preserved image tiles. The edge set $\mathcal{E}$ represents the bordering relationships between image tiles in an 8-adjacent manner as shown in Figure 2. Then, the generated graph $\mathcal{G}$ is able to depict the feature and spatial relations of the WSI, and is thus suitable for further analysis.

Following the principle of a Graph-Transformer architecture, our framework first uses a Graph Convolution (GC) layer to extract the task-shared low-level representation of $\mathcal{G}$. As illustrated in previous works (Wu et al. 2021; Rong et al. 2020b), the GNN part in Graph-Transformer architecture learns the representation at graph nodes from neighborhood features. This neighborhood aggregation of GNN is helpful for learning local and short-range correlations of graph nodes, and thus suitable to serve as the shared layers for multiple different tasks. The message propagation and aggregation of the graph are defined as

$$H_{l+1} = ReLU\left(\hat{A}H_l W_l\right), \quad l = 1, 2, \ldots, L. \tag{1}$$

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{2}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix A of graph $\mathcal{G}$ with added self-connections, and I_N is the identity matrix. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is a diagonal matrix, and $W_l \in \mathbb{R}^{d\times d}$ is a layer-specific trainable weight matrix. $H_l \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the input of the l_{th} GC layer, where $|\mathcal{V}|$ is the number of nodes and d is the dimension of each node, and H_l is initialized with the node features of $\mathcal{G}$.

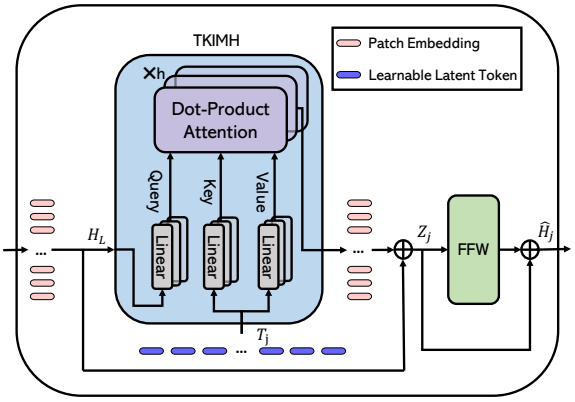

Figure 3: Illustration of Task-aware Knowledge Injection.

Task-aware Knowledge Injection

For a more accurate representation learning for different tasks, we propose a Task-aware Knowledge Injection module to store the task-specific knowledge in different task branches and thus transfer the task-shared feature from the task-shared GCN into task-specific feature spaces. The developed module calculates the correlation among the task-shared features with the task-specific knowledge based on the multi-head attention mechanism (Vaswani et al. 2017):

$$\mathrm{MH}(Q,K,V) = [O_1,\ldots,O_h]\,W^O, \tag{3}$$

$$O_i = \mathrm{Att}\left(QW_i^Q, KW_i^K, VW_i^V\right), \tag{4}$$

where $\mathrm{Att}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \sigma\left(\boldsymbol{Q}\boldsymbol{K}^T\right)\boldsymbol{V}$, h is the number of parallel attention layers, and σ is an activation function.

To transfer the task-shared features into task-specific spaces, we design a novel Task-aware Knowledge Injection multi-head cross attention (TKIMH) block, as shown in Figure 3. Specifically, we take the task-shared hidden representation H_{L} as the query ($\boldsymbol{Q}$), and the task-specific trainable latent tokens T_j as the key ($\boldsymbol{K}$) and value ($\boldsymbol{V}$) for the cross multi-head attention calculation. Each task branch has an independent set of trainable latent tokens, which is able to store the task-aware knowledge learned from the dataset during the training process. The TKIMH for task j can be denoted as:

$$\mathrm{TKIMH}(H_{\mathrm{L}},T_j) = [O_{1j},\ldots,O_{hj}]\,W_j^O, \tag{5}$$

$$O_{ij} = \mathrm{Att}\left(H_{\mathrm{L}}W_{ij}^Q, T_jW_{ij}^K, T_jW_{ij}^V\right), \tag{6}$$

where $H_{\mathrm{L}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the task-shared hidden representation, $T_j \in \mathbb{R}^{m\times d}$ is the learnable latent tokens containing the task-specific knowledge for task j, m is the number of the latent tokens, $W_{ij}^Q, W_{ij}^K, W_{ij}^V, W_j^O \in \mathbb{R}^{d\times d}$ are parameter matrices for linear projection operations for task j. Using the ingredients above, the Task-aware Knowledge Injection module for task j is defined as follows:

$$Z_j = \mathrm{LN}(H_{\mathrm{L}} + \mathrm{TKIMH}(H_{\mathrm{L}},T_j)), \tag{7}$$

$$\hat{H}_j = \mathrm{LN}(Z_j + \mathrm{rFF}(Z_j)), \tag{8}$$

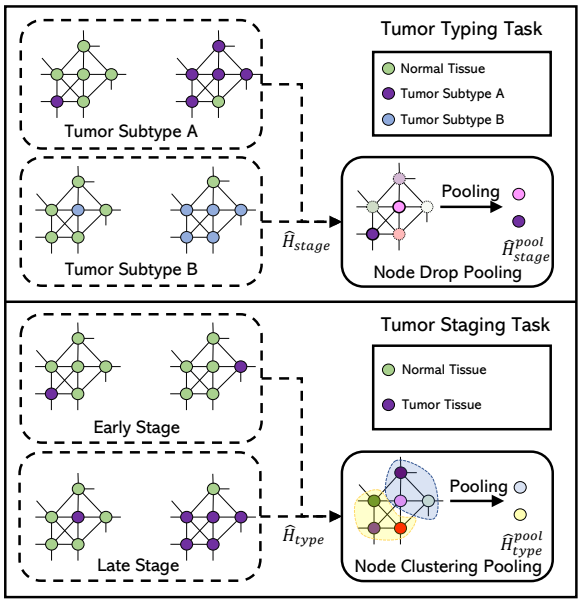

Figure 4: Overview of Domain Knowledge-driven Pooling.

where rFF is a row-wise feedforward layer that processes each individual row independently and identically, LN is a layer normalization (Ba, Kiros, and Hinton 2016), and $\hat{H}_j$ is the transferred task-specific graph features for task j. Note that the above module can be easily adapted to any Graph-Transformer architecture.

Domain Knowledge-driven Graph Pooling

Domain Knowledge-driven Graph Pooling is developed by task-aware pooling methods to meet the requirements of different downstream tasks. As shown in Figure 4, we adopt two different graph pooling methods (*i.e.*node drop method and node clustering method) for two tasks (*i.e.*tumor staging and tumor typing) with different diagnosis patterns.

Node Drop Pooling for Typing. During the clinical diagnosis process, the pathologists first examine the WSI to locate the tumor region and then determine the tumor type. Our node drop pooling method is designed to leverage the clinical process (as shown at the top of Figure 4). The model decision highly depends on the discriminative nodes (*i.e.*, tumor subtype A/B node) instead of the ratio or shape of different types of nodes in the whole graph. Therefore, the node drop method will be sufficient for the tumor typing task as long as one of the tumor nodes can be preserved. As previous work (Papp et al. 2021) has also pointed out that random dropping will increase the expressiveness of GNN, we implemented a random and independent node dropping in each training runs to generate the task-aware pooled representation $\hat{H}^{pool}_{type}$ for tumor typing task. Compared with the rank-based dropping method, our scheme will make the task more challenging and serve as a data augmentation method, which will make the corresponding branch more robust and more powerful in detecting discriminative image patches.

Node Clustering Pooling for Staging. Several elements influence the tumor stage diagnosis results of pathologists, including abnormal cells, the presence and size of tumor regions, and metastatic tumors. In general, the ratio and the shape of the tumor tissue nodes in the whole graph will be essential for the slide-level tumor stage diagnosis, as observed in the bottom of Figure 4. Node clustering pooling methods are more suitable for the staging task to preserve the whole graph information, as node drop methods may lose the above information during dropping. Inspired by GMPool (2021), We design GCMinCut, an improved version of MinCut Pooling (Bianchi, Grattarola, and Alippi 2020), in which we replaced the MLP during the pooling with an additional GC layer to import the neighboring information of the graph. The GCMinCut pooling can be denoted as follows:

$$\mathbf{S} = ReLU\left(\hat{A}\hat{H}_{stage}W_{\text{pool}}\right), \hat{H}^{\text{pool}}_{stage} = \mathbf{S}^T\hat{H}_{stage}, \quad (9)$$

where $\hat{H}_{stage} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the task-specific representation transferred by the task-aware knowledge injection module in the tumor staging branch, $W_{\text{pool}} \in \mathbb{R}^{d\times d}$ is a trainable weight matrix, and $\mathbf{S} \in \mathbb{R}^{p\times|\mathcal{V}|}$ is the assignment matrix for soft node clustering, p is the number of node clusters (*i.e.*, the number of nodes after pooling), and $\hat{H}^{pool}_{stage} \in \mathbb{R}^{p\times d}$ is the task-aware pooled representation for tumor staging task.

Technical Details and Training Procedure

After the Domain Knowledge-driven Graph Pooling module, the task-aware pooled representations are fed into a standard Transformer layer stack with no additive positional embeddings as the GNN has already encoded the structural information into the node embeddings. After that, we apply task-specific MLP heads for each branch to predict the task labels. The label prediction $\hat{Y}_i$ of task i can be denoted as:

$$\hat{X}_i = \text{Transformer}\left([CLS; \hat{H}^{\text{pool}}_i]\right), \hat{Y}_i = \text{MLP}\left(\hat{X}^{(0)}_i\right), \quad (10)$$

where $CLS \in \mathbb{R}^{1\times d}$ is the class token in Transformer.

To train the network, we first employed the cross-entropy loss for both tasks. Take the type prediction task as an example, the objectiveness is:

$$\mathcal{L}_{type} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C_{type}} Y^{(ij)}_{type}\log\left(\hat{Y}^{(ij)}_{type}\right), \quad (11)$$

where N is the whole sample number, C_{type} is category number for type prediction task, and Y is the one-hot label.

Then unsupervised MinCut pooling loss (Bianchi, Grattarola, and Alippi 2020) is adopted for extra regularization:

$$\mathcal{L}_{mincut} = \underbrace{-\frac{\text{Tr}\left(\mathbf{S}^T\tilde{\mathbf{A}}\mathbf{S}\right)}{\text{Tr}\left(\mathbf{S}^T\tilde{\mathbf{D}}\mathbf{S}\right)}}_{\mathcal{L}_c} + \underbrace{\left\|\frac{\mathbf{S}^T\mathbf{S}}{\|\mathbf{S}^T\mathbf{S}\|_F} - \frac{\mathbf{I}_p}{\sqrt{p}}\right\|_F}_{\mathcal{L}_o}, \quad (12)$$

where $\|\cdot\|F$ indicates the Frobenius norm. The cut loss term $\mathcal{L}_c$ encourages strongly connected nodes to be clustered together, and the orthogonality loss term $\mathcal{L}_o$ encourages the cluster assignments to be a similar size.

Finally, the total loss $\mathcal{L}_{total}$ can be denoted as the weighted summation of the above losses:

$$\mathcal{L}_{total} = w_t\mathcal{L}_{type} + w_s\mathcal{L}_{stage} + w_m\mathcal{L}_{mincut}. \quad (13)$$

Experiments

Datasets

We evaluate the proposed MulGT framework on two public datasets (KICA and ESCA) from The Cancer Genome Atlas (TCGA) repository. In the tumor staging task, patients with TNM labels as I/II stage are categorized as the early stage while patients with TNM labels as III/IV are categorized as the late stage. We excluded patients with missing diagnostic WSI, tumor type diagnosis, and TNM label. The details of the above two datasets are as follows:

- **KICA** is the kidney carcinoma project containing 371 cases with 279 early-stage cases and 92 late-stage cases. For the tumor typing task, there are 259 cases diagnosed as kidney renal papillary cell carcinoma and 112 cases diagnosed as kidney chromophobe.
- **ESCA** is the esophageal carcinoma project containing 161 cases with 96 early-stage cases and 65 late-stage cases. For the tumor typing task, there are 94 cases diagnosed as adenomas and adenocaricinomas and 67 cases diagnosed as squamous cell carcinoma.

Experimental Setup

The proposed framework was implemented using PyTorch (Paszke et al. 2019) and PyTorch Geometric (Fey and Lenssen 2019) frameworks. All experiments were conducted on a workstation with four NVIDIA RTX 3090 GPUs. For a fair comparison, the proposed framework and other SOTA methods were all tested using non-overlapping 512×512 image tiles cropped under $20\times$ magnification from the WSIs, we filtered out image tiles containing less than 85% tissue region. Besides, KimiaNet (Riasatian 2020) served as the feature extractor for all methods to convert each 512×512 image tile into 1024-dimensional features for graph initialization. We set the number of nodes after graph pooling as 100, following the setting in GT-MIL (2021). We select the number of latent tokens of Task-aware Knowledge Injection module as 150 by hyper-parameter searching. All methods are trained with a batch size of 8 for 40 epochs with the Adam optimizer. For evaluation, the area under the curve (AUC) of receiver operating characteristic, the accuracy (ACC), and the F1-score were adopted. All approaches were evaluated with five-fold cross-validation from three different runs (initializations).

Experimental Results

Comparison with State-of-the-art Methods. We compare our framework with eight single-task state-of-the-art (SOTA) methods for WSI analysis including: (1) ABMIL (2018), (2) Gated-ABMIL (2018), (3) CLAM-MIL (2021), (4) CLAM-SB (2021), (5) DeepAttn-MIL (2020), (6) DS-MIL (2021), (7) GT-MIL (2021), and (8) Trans-MIL (2022). For DS-MIL (2021) method, it was

Method	Typing			Staging		
	AUC	ACC	F1	AUC	ACC	F1
ABMIL (2018)	95.42 ± 2.02	89.90 ± 2.77	89.82 ± 2.75	75.35 ± 3.74	70.51 ± 1.88	58.54 ± 2.55
Gated-ABMIL (2018)	94.84 ± 1.60	88.63 ± 2.98	88.61 ± 2.87	73.65 ± 3.25	69.69 ± 2.33	58.65 ± 2.78
DeepAttnMIL (2020)	96.87 ± 1.44	91.37 ± 2.53	91.37 ± 2.49	76.53 ± 2.84	70.32 ± 2.19	58.44 ± 2.86
CLAM-MIL (2021)	84.93 ± 3.15	79.46 ± 2.91	78.57 ± 3.27	70.97 ± 3.20	70.32 ± 2.20	58.64 ± 3.17
CLAM-SB (2021)	95.69 ± 2.31	90.62 ± 2.93	90.60 ± 2.96	74.94 ± 4.22	70.39 ± 2.22	58.28 ± 2.78
DS-MIL (2021)	93.97 ± 2.52	87.08 ± 3.04	86.90 ± 3.12	73.21 ± 4.35	68.94 ± 2.37	59.31 ± 2.31
GT-MIL (2021)	97.20 ± 1.19	92.31 ± 2.52	92.33 ± 2.46	78.63 ± 3.56	71.20 ± 3.60	68.38 ± 3.37
Trans-MIL (2022)	95.56 ± 2.11	89.14 ± 3.30	89.04 ± 3.31	73.34 ± 3.15	68.56 ± 3.46	57.70 ± 2.34
Ours	$\mathbf{98.44 \pm 0.67}$	$\mathbf{93.89 \pm 1.60}$	$\mathbf{93.89 \pm 1.59}$	$\mathbf{80.22 \pm 1.94}$	$\mathbf{74.98 \pm 3.08}$	$\mathbf{72.55 \pm 2.48}$

Table 1: Comparison with other methods on KICA dataset. Top results are shown in bold.

Method	Typing			Staging		
	AUC	ACC	F1	AUC	ACC	F1
ABMIL (2018)	92.51 ± 3.39	86.47 ± 4.16	86.33 ± 4.23	53.01 ± 3.95	54.34 ± 3.02	51.36 ± 3.19
Gated-ABMIL (2018)	95.17 ± 2.47	88.54 ± 3.05	88.39 ± 3.12	50.64 ± 4.12	53.38 ± 4.22	53.54 ± 5.02
DeepAttnMIL (2020)	96.12 ± 1.84	90.64 ± 2.93	90.50 ± 3.07	61.87 ± 3.32	61.48 ± 4.28	50.59 ± 2.79
CLAM-MIL (2021)	77.89 ± 5.90	73.98 ± 5.25	73.55 ± 5.57	61.23 ± 4.15	58.38 ± 4.11	50.38 ± 4.14
CLAM-SB (2021)	95.85 ± 1.78	90.66 ± 3.02	90.57 ± 3.10	63.01 ± 3.05	59.96 ± 3.46	51.75 ± 3.79
DS-MIL (2021)	87.80 ± 3.97	81.63 ± 4.81	81.05 ± 5.16	61.75 ± 2.20	59.39 ± 3.30	54.36 ± 5.27
GT-MIL (2021)	95.93 ± 1.58	89.87 ± 3.64	89.83 ± 3.60	69.23 ± 3.64	65.20 ± 3.72	62.64 ± 3.22
Trans-MIL (2022)	94.24 ± 2.33	86.59 ± 3.17	86.48 ± 3.14	60.56 ± 4.72	61.47 ± 3.87	49.73 ± 3.32
Ours	$\mathbf{97.49 \pm 1.46}$	$\mathbf{92.81 \pm 2.35}$	$\mathbf{92.74 \pm 2.41}$	$\mathbf{71.48 \pm 3.42}$	$\mathbf{66.63 \pm 3.14}$	$\mathbf{65.73 \pm 2.83}$

Table 2: Comparison with other methods on ESCA dataset. Top results are shown in bold.

introduced with a pyramidal fusion mechanism for multi-scale WSI features. We only test its performance under the single-scale setting for a fair comparison. The results for KICA and ESCA datasets are summarized in Table 1 and Table 2, respectively. Overall, across all tasks and different datasets, our frameworks consistently achieve the highest performance on all the evaluation metrics. GT-MIL (2021) performs best among the previous SOTAs, which demonstrates the powerful representation of Graph-Transformer architecture in WSI analysis. However, compared with our methods, GT-MIL (2021) only built a Graph-Transformer network in the single-task setting and only adopted MinCut pooling (Bianchi, Grattarola, and Alippi 2020). In comparison with GT-MIL (2021), for instance, our framework achieved a performance increase of **1.24%** in AUC, **1.58%** in ACC, **1.56%** in F1-score for tumor typing task, and **1.59%** in AUC, **3.78%** in ACC, **4.17%** in F1-score for tumor staging task on KICA dataset, which validates the effectiveness of the designs in our framework. More importantly, we observe an obvious improvement in tumor staging tasks, which is more challenging among the two tasks. The reason is probably the more general and robust task-shared representations learned in the multi-task learning paradigm.

Ablation Study. We conducted an ablation study on KICA dataset to demonstrate the effectiveness of each proposed component. We first compare our Domain Knowledge-driven Graph Pooling module with node drop based methods as well as node clustering based methods. We test multiple node drop based methods (SortPool (2018), TopKPool (2019), and SAGPool (2019)) and node clustering based methods (DiffPool (2018), MinCutPool (2020), and GMPool (2021)), and report the best performance in the above two groups. As observed from the first three rows in Table 3, obvious improvement could be seen in all evaluation metrics except F1 in tumor staging, which demonstrates the effectiveness of exploiting the domain knowledge for different tasks during the graph pooling process. The effectiveness of the proposed TK-Injection module is shown by comparison with baselines with no task-specific transferring and simple task-specific linear projections from the last three rows in Table 3. Compared with baselines without task-specific transferring, performance increases can be found in all the evaluation metrics except AUC and ACC in tumor staging in task-specific linear projections, which demonstrate that it is essential to transfer the task-agnostic feature into different task-specific spaces during multi-task learning. Moreover, our cross-attention based task-ware knowledge injection module performs better in all aspects than task-specific linear projections, which illustrate the effectiveness of storing the task-specific knowledge in latent tokens and importing them via the attention mechanism.

Investigation of Multi-task Learning Paradigm. To figure out the effectiveness of the multi-task learning paradigm, we also tested our framework and the elaborately designed modules under the single-task setting on KICA dataset. Table 4 summarizes the experimental results, where the multi-task paradigm benefited both tasks, especially the more challenging one, *i.e.*, tumor staging. The ACC and F1-score in tumor staging task increase by 1.31% and 1.10%, respectively. The performance improvement in tumor typing is less

DomainPool	TK-Injection	Typing			Staging		
		AUC	ACC	F1	AUC	ACC	F1
Drop-based		95.72 ± 1.64	90.11 ± 2.39	90.01 ± 2.35	77.07 ± 2.33	71.78 ± 2.78	69.93 ± 3.70
Cluster-based		97.40 ± 0.99	91.97 ± 2.27	92.02 ± 2.13	80.12 ± 3.51	73.45 ± 2.82	71.47 ± 3.68
✓		97.90 ± 1.32	93.50 ± 1.91	93.53 ± 1.88	$\mathbf{80.67 \pm 3.51}$	74.07 ± 3.53	70.78 ± 3.33
✓	Linear	98.10 ± 0.55	93.59 ± 1.56	93.55 ± 1.61	79.86 ± 2.20	73.57 ± 2.94	71.06 ± 3.13
✓	✓	$\mathbf{98.44 \pm 0.67}$	$\mathbf{93.89 \pm 1.60}$	$\mathbf{93.89 \pm 1.59}$	80.22 ± 1.94	$\mathbf{74.98 \pm 3.08}$	$\mathbf{72.55 \pm 2.48}$

Table 3: Ablation study on KICA dataset. DomainPool and TK-Injection denote the Domain Knowledge-driven Graph Pooling module and the Task-aware Knowledge Injection module, respectively. Drop-based and Cluster-based denote that replacing the DomainPool with node drop pooling methods or node clustering pooling methods, respectively. Linear denotes that replacing the cross-attention mechanism in the Task-aware Knowledge Injection module with task-specific linear projections.

Paradigm	Typing			Staging		
	AUC	ACC	F1	AUC	ACC	F1
Single	98.41	93.07	93.08	80.14	73.67	71.45
Multi	**98.44**	**93.89**	**93.89**	**80.22**	**74.98**	**72.55**

Table 4: Comparison of single-task and multi-task paradigm on MulGT. Single for single-task while Multi for multi-task.

Scheme	Typing			Staging		
	AUC	ACC	F1	AUC	ACC	F1
Shared	98.23	93.30	92.32	78.54	72.71	69.28
Specific	**98.44**	**93.89**	**93.89**	**80.22**	**74.98**	**72.55**

Table 5: Comparison of different task-knowledge latent token schemes in MulGT. Shared means using a shared set of latent tokens, while Specific means using independent sets of latent tokens in different task branches.

than staging, as it has already achieved very high performance. However, note that our framework still outperforms all previous SOTAs in Table 1 including the GT-MIL (2021) under the single-task setting, which means that our Task-aware Knowledge Injection and Domain Knowledge-driven Pooling modules could also improve the performance of single-task based methods.

Investigation of Task-knowledge Latent Token. We also investigate the impact of different schemes for task-aware knowledge latent tokens on KICA dataset, and show the mean results in Table 5. The "Shared" scheme means that different task branches use a shared set of knowledge latent tokens, while the "Specific" scheme means that different task branches have independent knowledge latent token sets. The "Specific" scheme achieves better performance in all metrics, especially in F1-score for typing task and ACC and F1-score in staging task. These experimental results show that different diagnostic tasks require different knowledge (sets), which is the same as the pathologists' experiences.

Visualization of Task-specific Feature Spaces. We further conduct t-SNE visualization of two different WSIs to demonstrate the learned task-specific features in Figure 5. The blue and orange nodes denote transferred node features

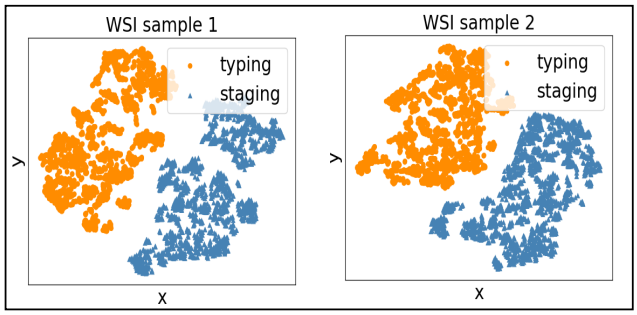

Figure 5: t-SNE visualization of task-specific features in different Task-aware Knowledge Injection modules.

in the Task-aware Knowledge Injection modules of typing branch and staging branch, respectively. All samples show a clear separation of the nodes with different colors, which means the task-shared features are indeed successfully transferred into different task-specific feature spaces.

Conclusion

In this paper, we propose a novel MulGT framework for WSI analysis with multi-task learning. By exploring the commonalities and different diagnosis patterns in different WSI diagnosis tasks, our framework is able to learn more general and robust task-shared representation as well as more accurate task-specific features. Specially, a Task-aware Knowledge Injection module is introduced to store and import the knowledge of different tasks, thus transferring the task-shared representation into different task-specific feature spaces. Meanwhile, to leverage the domain knowledge from the pathologists, a Domain Knowledge-driven Graph Pooling module is elaborately designed to simulate the diagnosis pattern of different analysis tasks. Above building commons lead to performance improvement on both tasks. Extensive experiments validate the prominence of the proposed framework. In the future, we will extend our framework to other WSI analysis tasks, such as survival prediction and prognosis analysis, with domain knowledge from pathologists. Meanwhile, we will develop a hierarchical multi-task Graph-Transformer framework to leverage the natural image pyramid structure of WSI for multi-scale analysis.

Acknowledgements

The work described in this paper was supported in part by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. T45-401/22-N) and in part by HKU Seed Fund for Basic Research (Project No. 202009185079 and 202111159073). The computations in this paper were partly performed using research computing facilities offered by Information Technology Services, The University of Hong Kong.

References

Amores, J. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence*, 201: 81–105.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baek, J.; Kang, M.; and Hwang, S. J. 2021. Accurate learning of graph representations with graph multiset pooling. *arXiv preprint arXiv:2102.11533*.

Bianchi, F. M.; Grattarola, D.; and Alippi, C. 2020. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, 874–883. PMLR.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chen, R. J.; Chen, C.; Li, Y.; Chen, T. Y.; Trister, A. D.; Krishnan, R. G.; and Mahmood, F. 2022. Scaling Vision Transformers to Gigapixel Images via Hierarchical Self-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16144–16155.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, 794–803. PMLR.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5177–5186.

Coudray, N.; Ocampo, P. S.; Sakellaropoulos, T.; Narula, N.; Snuderl, M.; Fenyö, D.; Moreira, A. L.; Razavian, N.; and Tsirigos, A. 2018. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10): 1559–1567.

Crawshaw, M. 2020. Multi-Task Learning with Deep Neural Networks: A Survey. *CoRR*, abs/2009.09796.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

Gao, H.; and Ji, S. 2019. Graph u-nets. In *international conference on machine learning*, 2083–2092. PMLR.

Guan, Y.; Zhang, J.; Tian, K.; Yang, S.; Dong, P.; Xiang, J.; Yang, W.; Huang, J.; Zhang, Y.; and Han, X. 2022. Node-Aligned Graph Convolutional Network for Whole-Slide Image Representation and Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18813–18823.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, W.; Yu, L.; Lin, C.; Huang, H.; Yu, R.; Qin, J.; and Wang, L. 2022. H2-MIL: Exploring Hierarchical Representation with Heterogeneous Multiple Instance Learning for Whole Slide Image Analysis. In *Proceedings of the AAAI conference on artificial intelligence*, 933–941. AAAI Press.

Ilse, M.; Tomczak, J.; and Welling, M. 2018. Attention-based deep multiple instance learning. In *International conference on machine learning*, 2127–2136. PMLR.

Javaloy, A.; and Valera, I. 2021. RotoGrad: Gradient Homogenization in Multitask Learning. *arXiv preprint arXiv:2103.02631*.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Khened, M.; Kori, A.; Rajkumar, H.; Krishnamurthi, G.; and Srinivasan, B. 2021. A generalized deep learning framework for whole-slide image segmentation and analysis. *Scientific reports*, 11(1): 1–14.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention graph pooling. In *International conference on machine learning*, 3734–3743. PMLR.

Li, B.; Li, Y.; and Eliceiri, K. W. 2021. 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*, 14318–14328.

Lin, K.; Wang, L.; and Liu, Z. 2021. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12939–12948.

Liu, B.; Liu, X.; Jin, X.; Stone, P.; and Liu, Q. 2021. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 18878–18890.

Lu, M. Y.; Williamson, D. F.; Chen, T. Y.; Chen, R. J.; Barbieri, M.; and Mahmood, F. 2021. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6): 555–570.

Maron, O.; and Lozano-Pérez, T. 1998. Attention is all you need. In *Advances in neural information processing systems*, 570–576.

Mialon, G.; Chen, D.; Selosse, M.; and Mairal, J. 2021. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*.

Min, E.; Chen, R.; Bian, Y.; Xu, T.; Zhao, K.; Huang, W.; Zhao, P.; Huang, J.; Ananiadou, S.; and Rong, Y. 2022. Transformer for Graphs: An Overview from Architecture Perspective. *arXiv preprint arXiv:2202.08455*.

Murthy, V.; Hou, L.; Samaras, D.; Kurc, T. M.; and Saltz, J. H. 2017. Center-focusing multi-task CNN with injected

features for classification of glioma nuclear images. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 834–841. IEEE.

Papp, P. A.; Martinkus, K.; Faber, L.; and Wattenhofer, R. 2021. Dropgnn: random dropouts increase the expressiveness of graph neural networks. *Advances in Neural Information Processing Systems*, 34: 21997–22009.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pataki, B. Á.; Olar, A.; Ribli, D.; Pesti, A.; Kontsek, E.; Gyöngyösi, B.; Bilecz, Á.; Kovács, T.; Kovács, K. A.; Kramer, Z.; et al. 2022. HunCRC: annotated pathological slides to enhance deep learning applications in colorectal cancer screening. *Scientific Data*, 9(1): 1–7.

Riasatian, A. 2020. *Kimianet: Training a deep network for histopathology using high-cellularity*. Master's thesis, University of Waterloo.

Riasatian, A.; Babaie, M.; Maleki, D.; Kalra, S.; Valipour, M.; Hemati, S.; Zaveri, M.; Safarpoor, A.; Shafiei, S.; Afshari, M.; et al. 2021. Fine-Tuning and training of densenet for histopathology image representation using TCGA diagnostic slides. *Medical Image Analysis*, 70: 102032.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; and Huang, J. 2020a. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33: 12559–12571.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; WEI, Y.; Huang, W.; and Huang, J. 2020b. Self-Supervised Graph Transformer on Large-Scale Molecular Data. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 12559–12571. Curran Associates, Inc.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vuong, T. L. T.; Lee, D.; Kwak, J. T.; and Kim, K. 2020. Multi-task deep learning for colon cancer grading. In *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–2. IEEE.

Wang, S.; Yang, D. M.; Rong, R.; Zhan, X.; and Xiao, G. 2019a. Pathology image analysis using segmentation deep learning algorithms. *The American journal of pathology*, 189(9): 1686–1698.

Wang, X.; Chen, H.; Gan, C.; Lin, H.; Dou, Q.; Tsougenis, E.; Huang, Q.; Cai, M.; and Heng, P.-A. 2019b. Weakly supervised deep learning for whole slide lung cancer image analysis. *IEEE transactions on cybernetics*, 50(9): 3950–3962.

Wang, Z.; Yu, L.; Ding, X.; Liao, X.; and Wang, L. 2022. Lymph Node Metastasis Prediction from Whole Slide Images with Transformer-guided Multi-instance Learning and Knowledge Transfer. *IEEE Transactions on Medical Imaging*.

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. *Advances in Neural Information Processing Systems*, 34: 13266–13279.

Yang, Y.; Yang, Y.; Yuan, Y.; Zheng, J.; and Zhongxi, Z. 2020. Detecting helicobacter pylori in whole slide images via weakly supervised multi-task learning. *Multimedia Tools and Applications*, 79(35): 26787–26815.

Yao, J.; Zhu, X.; Jonnagaddala, J.; Hawkins, N.; and Huang, J. 2020. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65: 101789.

Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31.

Zhang, J.; Zhang, H.; Xia, C.; and Sun, L. 2020. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*.

Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 4438–4445.

Zheng, Y.; Gindra, R.; Betke, M.; Beane, J.; and Kolachalama, V. B. 2021. A deep learning based graph-transformer for whole slide image classification. *medRxiv*.