Slide-based Graph Collaborative Training for Histopathology Whole Slide Image Analysis

Jun Shi, *Member, IEEE*, Tong Shu, Zhiguo Jiang, Wei Wang, Haibo Wu, and Yushan Zheng, *Member, IEEE*

***Abstract*— The development of computational pathology lies in the consensus that pathological characteristics of tumors are significant guidance for cancer diagnostics. Most existing research focuses on the inner-contextual information within each WSI yet ignores the possible inter-correlations between slides. As the development of tumors is a continuous process involving a series of histological, morphological, and genetic changes that accumulate over time, the similarities and differences between WSIs across various stages, grades, locations and patients should potentially contribute to the representation of WSIs and deserve to be taken into account in WSI modeling. To verify the advancement of introducing the slide inter-correlations into the representation learning of WSIs, we proposed a generic WSI analysis pipeline SlideGCD that can be adapted to any existing Multiple Instance Learning (MIL) frameworks and improve their performance. With the new paradigm, the prior knowledge of cancer development can participate in the end-to-end workflow, which concurrently initializes and refines the slide representation, as a guide for message passing in the slide-based graph. Extensive comparisons and experiments are conducted to validate the effectiveness and robustness of the proposed pipeline across 4 different tasks, including cancer subtyping, cancer staging, survival prediction, and gene mutation prediction, with 8 representative SOTA WSI analysis frameworks as backbones. The code is available at https://github.com/HFUT-miaLab/SlideGCD**

***Index Terms*— computer-aided diagnosis, computational pathology, graph learning, whole slide image classification.**

This work was partly supported by the Anhui Provincial Natural Science Foundation (Grant No. 2408085MF162), partly supported by the Beijing Natural Science Foundation (Grant No. 7242270), partly supported by the National Natural Science Foundation of China (Grant No. 61906058, 61901018, 62171007), partly supported by the Fundamental Research Funds for the Central Universities of China (Grant No. YWF-23-Q-1075), partly supported by Emergency Key Program of Guangzhou Laboratory (Grant No. EKPG21-32), partly supported by Joint Fund for Medical Artificial Intelligence (Grant No. MAI2023C014), and partly supported by National Key Research and Development Program of China (Grant No. 2021YFF1201000) and partly supported by Anhui Provincial Health and Medical Research Project (Grant No. AHWJ2023A10143). (Corresponding author: Yushan Zheng, Haibo Wu.)

Jun Shi is with the School of Software, Hefei University of Technology, Hefei 230601, China (e-mail:juns@hfut.edu.cn).

Tong Shu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China.

Zhiguo Jiang is with the Image Processing Center, School of Astronautics, Beihang University, Beijing, 102206, China and Tianmushan Laboratory, Beihang University, Hangzhou, 311115, Zhejiang, China.

Wei Wang and Haibo Wu are with the Department of Pathology, the First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230036, China and also with the Intelligent Pathology Institute, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230036, China (e-mail: wuhaibo@ustc.edu.cn).

Yushan Zheng is with the School of Engineering Medicine, Beijing Advanced Innovation Center on Biomedical Engineering, Beihang University, Beijing 100191, China (e-mail: yszheng@buaa.edu.cn).

I. Introduction

HISTOPATHOLOGICAL characteristics of tumors, including the tendency of tissue invasion, metastasis, growth pattern, etc., have been proven to effectively guide cancer diagnosis and therapies by numerous studies and practices [1]. Currently, whole slide images (WSIs) have been closely involved in medical practice as an indispensable part of the routine diagnostic process, becoming the gold standard for cancer diagnosis. In recent years, a large amount of research has focused on using artificial intelligence (AI) technology, especially deep learning, to examine WSIs and assist pathologists in effective, accurate, and reproducible pathological analysis and diagnosis, and has achieved significant accomplishments in various fields, e.g. cancer subtyping [2], [3], cancer staging [4], [5], survival prediction [6], [7], gene mutation prediction [8], [9], etc.

Considering the special attributes (the giga-pixel resolution and the pyramid structure) that distinguish WSIs from natural scene images, the current WSI analysis framework follows the Multiple Instance Learning (MIL) paradigm which takes patches as the smallest instance of analysis and explores the inner-contextual information of WSI by modeling the correlation between patches. Patch-based WSI analysis methods focus on how to model the relationships between patches more comprehensively and efficiently, and it can be divided into the following four categories: 1) Classical MIL methods [1], [10] treat each patch as an independent instance and generate slide-level representation by aggregating patch-level embeddings via different pooling methods. 2) A series of pseudo-bag-based methods [2], [11] has been proposed which divide the patches of each WSI into many separated pseudo-bags for solving data scarcity of annotated WSIs. 3) The graph-based methods [5], [9], [12]–[15] utilize the patch-based graph where patches are nodes and edges indicate the potential connections between them to simulate the relationships between patches and to represent WSI. 4) The sequence-based methods [16]–[18] consider WSI as a sequence of patches and involve

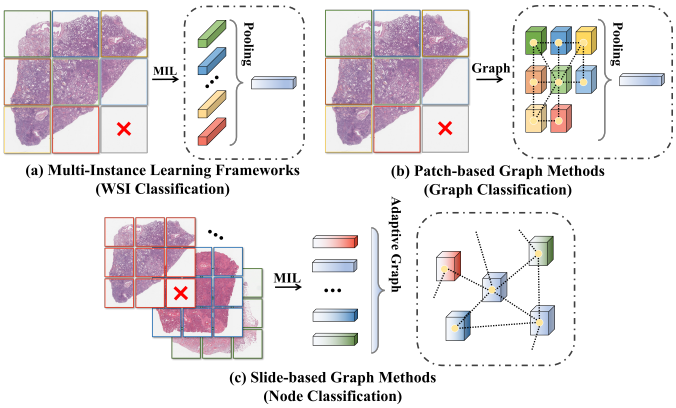

Fig. 1: Motivation of our method. (a) Multi-Instance Learning Frameworks. The main difference between MIL methods lies in the implementations of pooling operations. (b) Patch-based Graph Methods. The mainstream graph-based methods represent WSIs to graphs and transfer the WSI classification problem as graph classification. (c) Slide-based Graph Methods. SlideGCD conceptualizes the WSI classification problem as node classification to explore the inter-correlations between slides explicitly via GNNs.

various mechanisms or modules, e.g. Transformer or Structured State Space Models, to construct the detailed correlation among patches. With great achievements made by the above studies, the intra-relationships among patches from the same magnification are well-explored, and the interactions across magnifications are arousing more and more attention in recent years [3], [4], [19].

Although the inner-contextual information of WSIs is well-delved by previous research, the inter-correlations between WSIs have not drawn much attention. As most tumors develop through a continuous process involving a series of histological [20], morphological [21] and genetic changes [22] that accumulate over time, the similarities and differences between WSIs that across various stages, grades, locations and patients should potentially contribute to the analysis of WSIs and deserve to be taken into account for attaining better slide representations. Some studies [6], [14], [23] are aware of the importance of the inter-correlations in patch-slide-patient hierarchy yet stop within the patient level.

In this paper, we explore inter-correlations between WSIs on a larger scope and find an efficient way to unite inter-correlations with the intra-correlations in each WSI. Specifically, we propose the **S**lide-based **G**raph **C**ollaborative training pipeline with knowledge **D**istillation (SlideGCD) for WSI representation learning that dynamically organizes the slide embeddings into a slide-based graph and makes message passing between connected slides via graph neural networks. The intuitive differences between the patch-based graph and the slide-based graph are shown in Fig. 1. More concretely, we take existing MIL methods as the backbone to obtain the initial slide-level embeddings. Then, SlideGCD is used to explore the contextual information implied in the extensive slide-based graph. Finally, the slide-level predictions are obtained by conducting node classification on the slide-based graph.

The main contributions of this paper are summarized below:

- We propose a new histopathology WSI analysis paradigm that involves prior knowledge of cancer development with coordinatively constructing the slide-based graph and conducting graph message passing during the representation learning.
- We devise a rehearsal-based adaptive graph construction strategy to model the slide-level inter-correlations. Besides, a knowledge distillation (KD) based collaborative training for the slide-based hypergraph convolutional network is applied to transfer and enhance the intra-contextual information learned by the MIL network.
- We extensively evaluate our method by applying it to 8 representative SOTA WSI frameworks, demonstrating consistent effectiveness and generalization across 4 downstream tasks and 4 feature encoders under varying training schemes, while offering empirical insights for diverse scenarios.

II. Related Works

This section reviews the approaches related to graph-based WSI analysis and the methods that potentially explore the slide-level inter-correlations.

A. Graph-based WSI Analysis

Due to its flexibility and interpretability, much attention has been put on the graph structure and graph neural networks. Graph-based methods have shown competitive performance on various WSI analysis tasks. According to different graph structures applied, existing methods can be grouped into the following three categories.

1) Methods on Regular graphs: On account of the analogy that WSI is the graph where its patches are the nodes, graph structures were introduced into WSI analysis at its very early stage. DeepGraphConv [12] randomly samples 1000+ patches and connects them with a feature similarity threshold to construct a patch-based graph for each WSI. PatchGCN [13] builds patch-based graphs via spatial adjacent patches and gains better performance on the survival prediction task. LAMIL [24] and GTP [25] follow the graph construction strategy of PatchGCN yet design different graph transformers to make efficient message passing. NAGCN [14] introduces the hierarchical global-to-local patch-based graph to represent WSI in both spatial and embedding space. SlideGraph+ [15] uses the biomarker attributes and neural network embeddings to build patch-based graphs and represent the complex organization of cells and the overall tissue micro-architecture. As the idea of multi-scale is progressively involved in WSI analysis, many graph-based methods have also kept up with this trend. SGMF [5] constructs the structure-aware hierarchical graph that considers tissue regions and patches from different magnifications in an interactive way. DAS-MIL [3] is another method that involves knowledge distillation, it creates patch-based graphs on various magnifications and utilizes the knowledge distillation mechanism to align the representation learned from different graphs. Although DAS-MIL and the proposed

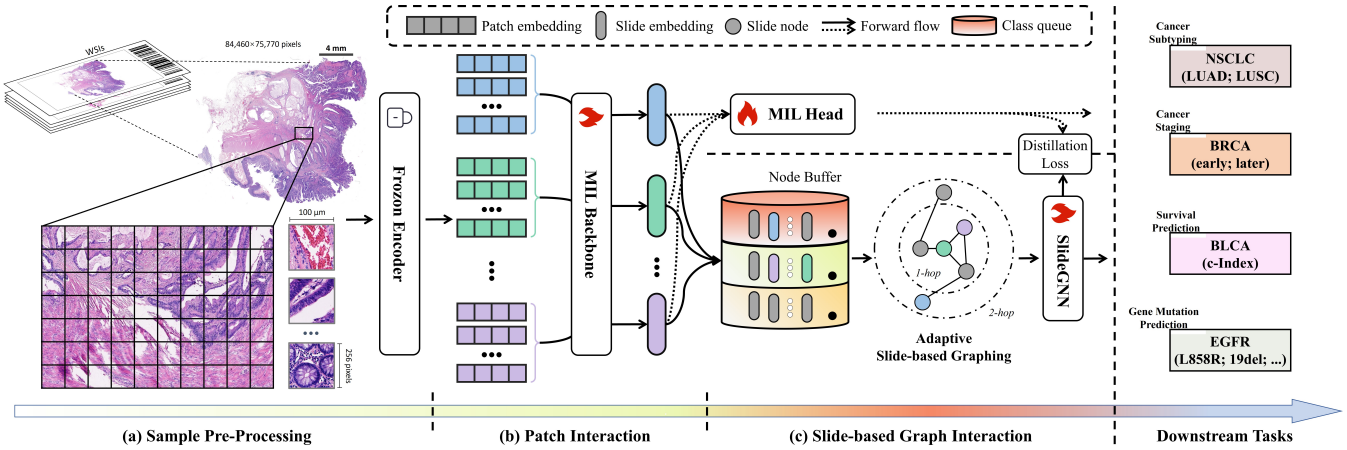

Fig. 2: Illustration of the proposed SlideGCD framework. The framework consists of three phases, (a) Sample Pre-Processing. Each WSI is transformed into a sequence of patch embeddings following the universal settings of the MIL paradigm. (b) Patch Interaction. Slide embeddings are generated by the backbone MIL method for each sample. (c) Slide-based Graph Interaction. A slide-based graph is maintained and updated during each mini-batch training (colored indicators symbolically denote samples from the current mini-batch), then graph learning and knowledge distillation are conducted to explore slide-level correlations and align both branches.

SlideGCD both use knowledge distillation techniques, DAS-MIL focuses on aligning the learned graph representations at different magnifications, while SlideGCD aims to transfer the inner-contextual knowledge learned by the original classifier to the inter-correlation-focused graph branch.

2) Methods on Graph-variant: Simple graphs consider all nodes equally and their connections can only describe pair-wise relationships. To deal with such drawbacks that deface representation learning in real-world applications, such as WSI representation, the heterogeneous graph and the hypergraph are engaged. HEAT [19] utilizes a heterogeneous graph with various pre-defined types of nodes to exploit the heterogeneity within WSI and perform WSI classification. Di et al. successively proposed two hypergraph-based WSI analysis frameworks b-HGFN [26] and HGSurvNet [27]. b-HGFN focuses on efficiently processing the hypergraphs with large-scale vertices, where hyperedges are constructed by adopting the K-nearest neighbor (k-NN) strategy on embedding space. HGSurvNet establishes the multi-hypergraph composed of topology-wise sub-hypergraph and phenotype-wise hypergraph for survival prediction with WSIs. MaskHGL [9] refines the hypergraph construction strategy with global alignment and designs a mask-reconstruction mechanism for achieving better performance on cancer subtyping and gene mutation prediction.

3) Methods on Adaptive graphs: Apart from the application of the variants of the graph in structure, the adaptive graph where its edge connections could be altered during training is getting attention as well. Hou et al. [28] design a dynamic structure learning module to assist the proposed spatial-hierarchical graph neural network in learning multi-scale information in WSIs. Liu et al. [29] develop a survival-aware structure learning module to construct the adaptive graph for global WSI representation calculation along with the fixed initial graph. WiKG [30] conceptualizes WSIs as a form of knowledge graph structure and dynamically builds the edges via a knowledge-aware attention mechanism during training.

Compared with the previous patch-based graph methods mentioned above, this work exploringly molds the WSIs into a slide-based graph with a rehearsal-based adaptive graph construction module to exploit the slide-level inter-correlations implied in the continuous changes during tumor development.

B. Slide-level Inter-Correlation Exploration

In clinical practice, each patient probably has multiple WSIs. How to obtain more accurate diagnostic results through multiple slide-level predictions has become a concern for many colleagues. Fan et al. [6] propose an aggregation module that takes slide-level embeddings produced by the front MIL framework to generate patient-level prediction. HVTSurv [23] and P&SrE [31] choose the Transformer model for interaction between the patient and its belonging slides. The difference is that HVTSurv cascades three transformer blocks for patch-level, WSI-level, and patient-level interaction respectively, yet P&SrE considers that the patient-level embeddings and slide-level embeddings are equal and thus utilizes a single transformer block for their interaction. BSN [32] is a primitive bag-similarity-based multi-instance learning method that calculates the similarities between the target bag and the reference bags with the instance-level representations and then deploys pooling operations to aggregate the bag-level representation. HistoKernel [33] is a nearly proposed non-deep learning algorithm based on the Maximum Mean Discrepancy (MMD) kernel that measures the distributional similarity between WSIs. It computes the pairwise MMD kernel based on patch features to estimate the WSI similarity and leverages Support Vector Machines (SVMs) to solve prediction problems. Some prototype-based algorithms [34], [35] can be viewed as another type of slide-level interaction if we consider the prototypes

as some abstract slide representations. The similarity between naive WSIs and these fictional bags contributes a lot to their improvement. In addition, some other methods, although not intentionally focusing on the slide inter-correlation, potentially acknowledge the consistency among slides. NAGCN [14], HEAT [19] and MaskHGL [9] are aware that there should be a consistency between the constructed graphs in structure or node types. In their graph construction strategies, NAGCN and MaskHGL apply a hierarchical clustering method based on all the patches no matter which WSI it comes from to align the graph structure across slides. HEAT employs a pre-trained network to classify the patches into pre-defined node types thus achieving node-level consistency.

From the perspective of slide-level inter-correlation exploration, the above methods are not comprehensive enough, as they either only involve limited slide-level interactions (within a single case [23], [31] or between two slides [33], [34]) or do not model the slide-level inter-correlations in an explainable way [9], [14], [19], [32]. The proposed SlideGCD lifts the patient-level restriction and explicitly constructs a slide-based graph to explore the contextual information.

III. Methodology

In this section, we describe our proposed framework which consists of three phases: sample pre-processing, patch interaction, and slide-based graph interaction, as illustrated in Fig. 2. In the first phase, each WSI sample is processed into a sequence of patch embeddings with a pre-trained patch encoder. For the patch interaction phase, we engage the existing MIL method as the backbone to generate slide-level embeddings. In the slide-based graph interaction phase, a rehearsal-based adaptive graph construction strategy is exploited to build and maintain a slide-based graph during training. Then, the SlideGNN is deployed to explore the slide inter-correlation based on the slide-based graph and refine the slide embeddings for solving downstream tasks more precisely. Additionally, an online distillation is designed between the MIL head and the SlideGNN to solve the problem of knowledge misalignment.

A. Overall Workflow

We first summarize the overall workflow of the framework to provide a clear description of the end-to-end training process and actual inference process of SlideGCD. Regarding the training phase, there are several epochs of warmup (i.e. 10 epochs) at the beginning to make the MIL backbone pre-converge in advance so that the slide embeddings generated by the training model for the same sample would not have extreme fluctuations. In warmup epochs, only the MIL network, including the MIL backbone and the MIL head, is involved in forward and backward and the node buffer is updated with the First-In-First-Out (FIFO) strategy. After that, in the remaining training epochs, the graph branch is also involved and the Class-aware Node Buffer (CANB) begins to function as designed. At the inference stage, all parameters and the Class-aware Node Buffer are frozen. When a WSI is inputted, 1) its initial embedding will be made with the backbone, 2) the initial slide embedding will be inserted into the slide-based graph with the same rehearsal-based adaptive graph construction strategy, 3) the SlideGNN will make message passing to refine its embedding for final prediction.

B. Sample Pre-Processing

Assuming there is a dataset with N WSIs denoted as $\mathcal{D} = \{(\mathbf{B}_i, y_i)\}_{i=1}^{N}$. Each WSI $\mathbf{B}_i = \{I_{i,j}\}_{j=1}^{M_i}$ is annotated with a label $y_i \in \{0, ..., C-1\}$, where $I_{i,j}$ is tiled patch without patch-level label, M_i is the number of patches and C represents the number of categories. Then, there is a pre-trained patch encoder $f(\cdot)$, where we used PLIP [36], to transform the patch I_k into patch embeddings $\mathbf{x}_k \in \mathbb{R}^{512}$.

C. Patch Interaction

After obtaining the patch embeddings, an aggregator network is needed to exploit the patch correlations and generate slide embeddings. We leave this job to the "MIL Backbone", the key sub-network of existing MIL methods. Specifically, we divide the existing MIL framework into two parts based on the generation of slide representations that are ultimately used for downstream task prediction. The formal part that processes patch embeddings and generates slide-level representations is referred to as the MIL backbone. The later part, which makes the downstream task-related slide-level prediction $\hat{\mathbf{y}}^{MIL}$, is regarded as the MIL Head. In an ideal implementation, the backbone applied can come from any MIL method with any architecture as long as it can produce fixed D_s dimensional slide-level embeddings $\mathbf{s}_i \in \mathbb{R}^{D_s}$. Note that the slide embeddings are quite unstable during the first few training epochs. Such instability might defect the subsequent graph learning. Therefore, we set a few warmup epochs at the beginning of the training for backbone pre-convergence, in which only the MIL network is involved in forward and backward.

We employed several representative MIL methods as the backbone and conducted different downstream tasks for the proposed SlideGCD, more analysis can be found in Section IV.

D. Slide-based Graph Interaction

With the slide embeddings, the requirements for slide-based graph interaction have been fulfilled. In this section, we describe the main contributions including the rehearsal-based adaptive graph construction strategy, the details of the designed SlideGNN, and the collaborative training with knowledge distillation.

1) Rehearsal-based Adaptive Graph Construction: With the inspiration of the idea of the Memory Bank [37], [38] and the rehearsal-based continual learning [39], [40], a Class-aware Node Buffer is designed to store the previous slide embeddings which will participate in the slide-based graph construction and will be replayed in graph learning. Specifically, the Class-aware Node Buffer defines a storage space $\mathbf{Q} \in \mathbb{R}^{L \times D_s}$ that can deposit L slide embeddings. Given a downstream task with C categories, the buffer can be partitioned to C sub-queues $\mathbf{Q} = [\mathbf{Q}_1, ..., \mathbf{Q}_c, ..., \mathbf{Q}_C]$, where $\mathbf{Q}_c \in \mathbb{R}^{\frac{L}{C} \times D_s}$. Each sub-queue $\mathbf{Q}_c$ is responsible for storing slide embeddings with the

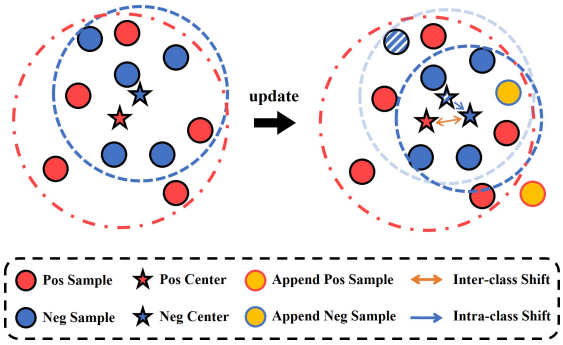

Fig. 3: Illustration of the update of the Node Buffer in embedded space. Left: the initial state of the Node Buffer after the warmup, where the centers of each category are close. Right: the updated state after a mini-batch with 2 samples. The appended positive sample is too far from the center even compared with the farthest stored node thus it won't be updated into the buffer since we consider it as an amplified disturbance from the former network update. The appended negative sample is close enough to the center to replace the farthest negative node marked by stripes. At last, the outcome of the update is that the negative center shifts away from the positive center as we expected.

Algorithm 1: the Pseudo-code of Node Buffer Update

Input: $\boldsymbol{x} \in \mathbb{R}^{D_s} \leftarrow$ The slide embedding to be stored;
$y \leftarrow$ The label of current slide embedding;
$\mathbf{Q} \in \mathbb{R}^{L \times D_s} \leftarrow$ The rehearsal-based node buffer;
$L \leftarrow$ The length of the node buffer;
$C \leftarrow$ The total number of categories.
Output: the updated node buffer $\mathbf{Q}'$

if in warmup stage **then**
  $\mathbf{Q}' \leftarrow concat([\boldsymbol{x}, \mathbf{Q}])[:L]$
else if in formal training stage **then**
  $\mathbf{Q}_t \leftarrow \mathbf{Q}.reshape((C, \frac{L}{C}, D_s))[y]$
  $center \leftarrow mean(\mathbf{Q}_t)$
  # $dist()$ is the function calculating the distance between inputs.
  $dists \leftarrow [dist(\mathbf{Q}_t[i], center)$ for i in range$(\frac{L}{C})]$
  $max\_value, max\_index = max(dists)$
  if $dist(\boldsymbol{x}, center) < max\_value$ **then**
    $\mathbf{Q}[y, index, :] = \boldsymbol{x}$
  end if
  $\mathbf{Q}' \leftarrow \mathbf{Q}.reshape((L, D_s))$
end if
return $\mathbf{Q}'$

c-th category. Each slide embedding stored in the buffer or the current mini-batch will correspond to a node in the subsequent slide-based graph and will be considered as its initial node embedding.

In addition, the rehearsal buffer will be updated during each mini-batch of training. During the warmup stage, we directly apply the First-In-First-Out (FIFO) strategy to update the node buffer, in which the newest mini-batch of slide embeddings will be randomly pushed into the node buffer and the outdated slide embeddings at the end of the buffer will be popped out simultaneously. In the formal training stage, we first calculate the centers of each sub-queue in the embedded space, and then each slide embedding in the current mini-batch is going to replace the farthest (e.g. measured by vector inner product) stored sample in the corresponding sub-queue as long as it is closer. The pseudo-code of the update strategy is shown in Algorithm 1. As updates continue, ultimately the sub-queues will be separated from each other. The following loss function is applied to ensure that during formal buffer updating.

$$L_{update} = -\sum_{\mathbf{u}\in\mathbf{U}} \log \frac{\exp(\mathbf{u}\cdot\mathbf{q}_+/\tau)}{\sum_{i=1}^{C}\exp(\mathbf{u}\cdot\mathbf{q}_i/\tau)} + \sum_{\mathbf{u}\in\mathbf{U}}\sum_{i=1}^{C} \mathbb{1}_{\mathbf{q}_i\neq\mathbf{q}_+}\left(\frac{\mathbf{u}\cdot\mathbf{q}_i}{\|\mathbf{u}\|\|\mathbf{q}_i\|}\right), \quad (1)$$

where $\mathbf{U}$ is the enqueued slide embeddings, $\mathbf{q}_i$ indicates the center of i-th sub-queue, the $\mathbf{q}_+$ represents the center which corresponds to the category of $\mathbf{u}$ and the τ is the temperature coefficient. $\mathbb{1}_{\mathbf{q}_i\neq\mathbf{q}_+}(\cdot)$ means that this term is not 0 only if $\mathbf{q}_i \neq \mathbf{q}_+$. An illustration of the update is shown in Fig. 3.

After getting the preliminary separable nodes, we conduct the adaptive graph generation (AGG) strategy to infer the inter-dependencies from the embedding space for connecting these nodes with hyperedges. Our AGG strategy consists of a linear layer that transforms the slide embeddings into an intermediate hidden space and a k-NN clustering operator that will be performed on the intermediate embeddings to connect each node with its k nearest neighbors with a hyperedge. Eventually, the slide embeddings in the current mini-batch and the slide embeddings retrieved from the node buffer are formulated as a hypergraph $\mathcal{G}$. The adaptive graph generation process can be formulated as:

$$\mathcal{G} = (\mathbf{X}^{(0)}, \mathcal{E}),\ \mathcal{E} = \mathrm{KNN}(\mathbf{P}, k),\ \mathbf{P} = \mathrm{Linear}(\mathbf{X}^{(0)}), \quad (2)$$

where $\mathbf{X}^{(0)} \in \mathbb{R}^{(L+B)\times D_s}$ is the node embeddings sequence that consists of the data in the buffer with a length of L and the current mini-batch data with a length of B (batch size) and $\mathcal{E}$ is the set of hyperedges that describes the connections between nodes.

2) Graph Learning via SlideGNN: With the slide-based graph representation, the SlideGNN composed of two hypergraph convolutional layers [42] and a Centering-Attention module is applied to explore the implied information, as:

$$\mathbf{X}^{(i+1)} = \mathrm{LeakyReLU}(\mathrm{HGC}(\mathbf{X}^{(i)}, \mathcal{E})), i \in \{0, 1\}, \quad (3)$$

$$\mathbf{H} = \mathrm{Concat}(\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}), \mathbf{H} \in \mathbb{R}^{L\times 3D_S}, \quad (4)$$

where $\mathrm{HGC}(\cdot)$ denotes the hypergraph convolution and $\mathbf{X}^{(i)}$ contains the information accumulated from the node itself to its i-hop neighbors.

Given that the adaptive graphs can involve graph heterophily that nodes with different categories are connected, we designed the Centering-Attention module to alleviate such heterophily by rebalancing the participation of k-hop information. The

Centering-Attention module is implemented with channel-wise attention and $Centering$ operation [43] which prevent the attention score from always being positive even when facing defective partial information. The computations can be formulated as:

$$\mathbf{H}' = \mathbf{H} \cdot Centering(\mathbf{a}) = \mathbf{H} \cdot (\mathbf{a} - \mathrm{Mean}(\mathbf{a})), \quad (5)$$

$$\mathbf{a} = \mathrm{Sigmoid}(\mathrm{ReLU}(\mathbf{H}^{\mathrm{T}}\mathbf{W}_0)\mathbf{W}_1), \quad (6)$$

where $\mathbf{W}_0, \mathbf{W}_1 \in \mathbb{R}^{L\times L}$ are the learnable weights, $\mathbf{a} \in \mathbb{R}^{3D_S}$ is the attention score before $Centering$. Finally, a linear classifier Cls_{Graph} with one linear layer is used to make final predictions $\hat{\mathbf{y}}^G$ for current mini-batch:

$$\hat{\mathbf{y}}^G = \mathrm{Softmax}(\mathrm{Linear}(\mathbf{H}')). \quad (7)$$

3) Collaborative Training with Knowledge Distillation: With the rehearsal-based adaptive graph and the SlideGNN, this framework is capable of exploiting the slide inter-correlations from the given WSI dataset. However, there is a knowledge misalignment between the graph branch and the MIL head ($\mathrm{Head}_{\mathrm{MIL}}$), in which the graph branch pays attention to exploring the slide-level inter-correlations, yet the MIL head focuses on digging slide intrinsic contextual information. To fully utilize the learned knowledge from both branches and align them, we introduced knowledge distillation to transfer the learned knowledge from $\mathrm{Head}_{\mathrm{MIL}}$ to the SlideGNN.

Technically, we treat the $\mathrm{Head}_{\mathrm{MIL}}$ and the SlideGNN as the teacher and student model separately, letting SlideGNN draw on the beneficial information learned by $\mathrm{Head}_{\mathrm{MIL}}$ with the Jensen-Shannon (JS) divergence loss [44], [45], as:

$$L_{KD} = \sum_{i=1}^{C} p_i^G \log\left(\frac{2p_i^G}{p_i^G + p_i^{MIL}}\right) + \sum_{i=1}^{C} p_i^{MIL} \log\left(\frac{2p_i^{MIL}}{p_i^G + p_i^{MIL}}\right),$$
$$p_i^G = softmax(\hat{y}_i^G, t), \quad p_i^{MIL} = softmax(\hat{y}_i^{MIL}, t), \quad (8)$$

where t is the temperature coefficient.

Then, the final loss of SlideGCD can be written as below, $L_{CE}(\cdot)$ represents the Cross-Entropy loss function, and β is the weight contributed by the buffer update:

$$L = L_{CE}(\hat{\mathbf{y}}^{MIL}, y) + L_{CE}(\hat{\mathbf{y}}^G, y) + L_{KD} + \beta L_{update}. \quad (9)$$

IV. Experiments

A. Datasets

We conducted extensive experiments on various downstream tasks to verify the effectiveness of the proposed pipeline, including cancer subtyping, cancer staging, survival prediction, and gene mutation prediction.

1) Cancer Subtyping: Two publicly available WSI datasets are used to evaluate our proposed pipeline in the downstream task of cancer subtyping, including the TCGA-BRCA and the TCGA-NSCLC released by The Cancer Genome Atlas (TCGA) project [46][1]. Specifically, TCGA-BRCA contains 998 diagnostic digital slides of two breast cancer subtypes, made up of 794 WSIs of invasive ductal carcinoma (IDC) and 204 WSIs of invasive lobular carcinoma (ILC). TCGA-NSCLC is a collection of two subtype projects for lung cancer, i.e. lung squamous cell carcinoma (LUSC) and lung adenocarcinoma (LUAD), for a total of 995 diagnostic WSIs, including 496 WSIs of LUSC and 499 WSIs of LUAD.

2) Cancer Staging: In the cancer staging task, we used the same WSIs with another set of labels for staging from the two public datasets mentioned above, TCGA-BRCA and TCGA-NSCLC. Concretely, we excluded samples without grading labels and categorized the remaining WSIs as early-stage (Stage I&II) and late-stage (Stage III&IV). In TCGA-BRCA, there are 713 WSIs of early-stage and 232 WSIs of late-stage. In TCGA-NSCLC, there are 726 WSIs of early-stage and 180 WSIs of late-stage. Note that the staging task is normally harder than subtyping because it relies more on subtle cellular morphological features and the natural class imbalance of its datasets.

3) Survival Prediction: As to the survival prediction task, we evaluated SlideGCD with the TCGA-BLCA (437 patients) and TCGA-UCEC (540 patients) cohorts, one diagnosis-related WSI is selected as the sample for each patient following the setting of previous studies [13].

4) Gene Mutation Prediction: An in-house clinical dataset USTC-EGFR[2] and a public dataset TCGA-EGFR are used to evaluate whether the slide inter-correlation can benefit the efficiency of gene mutation prediction via H&E histopathology WSIs. USTC-EGFR contains a total of 754 WSIs of lung histopathology from five categories of samples, including 165 WSIs of negative (Neg), 118 WSIs with a missense mutation in exon 21 (L858R), 184 WSIs with in-frame deletions in exon 19 (19del), 146 WSIs of wild type (Wild) and 141 WSIs with other driver gene mutations (Others). All labels have been confirmed by pathologists. TCGA-EGFR consists of 696 WSIs from 364 patients for EGFR mutation detection. Its genetic information is found in cBioPortal[3]. These WSIs are divided into two categories: wild type (Wild, 590 WSIs) and EGFR-mutant (Mutant, 106 WSIs).

B. Implementation Details

Before training, the 256×256 patches within tissue regions were split under 20× lenses with a standard resolution of 0.5 microns per pixel (MPP). During training, the Adam optimizer with Cosine Annealing learning rate scheduler was employed. For the sake of fairness, we set the batch size to 64 in all experiments, and if the method is unable to change its batch size from 1, we conduct a gradient accumulation with a step of 64. A maximum of 100 training epochs with early stopping is adopted. We directly adopted the default setting of hyper-parameters to the baseline from its public repository and remained fixed when applying SlideGCD except the learning rate will be reset to 1e-4 after the warmup for SlideGCD. All methods are implemented in Python with the *PyTorch* 1.8 and *PyTorch Geometric (PyG)* libraries. We ran the experiments on a computer with an NVIDIA RTX 3090 GPU.

[1]https://portgdc.cancer.gov/

[2]The study was approved by the Medical Research Ethics Committee of the First Affiliated Hospital of the University of Science and Technology of China (Anhui Provincial Hospital) under the protocol No.2022-RE-454.

[3]cBioPortal for Cancer Genomics: https://www.cbioportal.org/

TABLE I: Comparisons of the baselines and corresponding SlideGCD collaboration version on cancer subtyping. †: When reproducing HiGT, we applied the default setting on multi-scale from its original paper that only considered the thumbnail, 5×, and 10× magnifications. That is the reason why the performance of HiGT gaps with other methods.

Method	SlideGCD	TCGA-BRCA ACC(%)	TCGA-BRCA AUC(%)	TCGA-BRCA F1(%)	TCGA-NSCLC ACC(%)	TCGA-NSCLC AUC(%)	TCGA-NSCLC F1(%)
ABMIL [10]	✗	88.97±0.85	89.87±0.89	81.61±2.27	86.96±1.16	95.14±0.51	86.89±1.20
	✔	89.77±1.04	91.33±1.52	83.87±1.83	**91.04±2.42**	97.10±1.49	**91.02±2.45**
	Δ	+0.80	+1.46	+2.26	+4.08	+1.96	+4.13
CLAM [1]	✗	89.24±0.88	89.64±1.08	82.43±1.85	87.29±0.82	95.78±0.54	87.26±0.82
	✔	90.76±1.08	90.48±2.98	85.53±1.98	90.64±1.29	96.35±0.54	90.62±1.30
	Δ	+1.52	+0.84	+3.10	+3.35	+0.57	+3.36
PatchGCN [13]	✗	84.80±1.77	87.18±1.55	75.11±4.57	86.62±2.38	94.81±1.82	86.59±2.43
	✔	86.00±0.87	89.79±1.30	76.49±0.78	89.63±1.68	**97.62±0.75**	89.60±1.69
	Δ	+1.20	+2.61	+1.38	+3.01	+2.81	+3.01
TransMIL [16]	✗	88.17±1.00	90.99±0.91	82.09±1.75	85.82±1.67	94.82±1.17	85.77±1.67
	✔	90.70±1.73	92.82±2.00	85.91±2.62	90.70±3.00	96.53±1.22	90.68±3.03
	Δ	+2.53	+1.83	+3.82	+4.85	+1.71	+4.91
DTFDMIL [2]	✗	89.30±0.44	90.08±0.86	83.17±1.43	86.42±1.07	95.59±0.66	86.40±1.06
	✔	**91.50±0.50**	**92.83±0.93**	**86.52±0.78**	90.84±1.83	96.31±1.84	90.82±1.84
	Δ	+2.20	+2.75	+3.35	+4.42	+0.72	+4.42
HiGT † [4]	✗	86.09±1.13	86.20±0.59	77.93±1.14	85.42±1.96	93.93±0.42	85.33±2.06
	✔	88.43±0.40	87.93±2.07	81.47±0.34	87.32±1.53	94.65±0.58	87.28±1.56
	Δ	+2.34	+1.73	+3.54	+1.90	+0.72	+1.95
S4MIL [17]	✗	87.84±0.50	89.29±1.71	81.18±1.00	88.03±0.86	95.05±0.70	88.01±0.87
	✔	89.30±1.62	90.12±2.34	83.35±1.79	89.50±1.88	96.04±1.42	89.48±1.88
	Δ	+1.46	+0.83	+2.17	+1.47	+0.99	+1.47
MaskHGL [9]	✗	88.97±0.71	89.71±1.98	82.48±0.93	87.76±1.38	94.85±0.44	87.74±1.39
	✔	89.11±0.77	90.31±1.29	82.94±1.53	88.16±2.63	95.98±0.54	88.12±2.69
	Δ	+0.14	+0.60	+0.46	+0.40	+1.13	+0.38

All experiments are performed using the same five-fold cross-validation splits. Except for the survival prediction task, we randomly separate the datasets into train and test sets by the proportion of 7:3. For the TCGA-BLCA&UCEC, we follow the dataset partition in [13]. The average accuracy (ACC), macro-average F1 score (F1), and macro-average area under the receiver operating characteristic curve (AUC) are calculated for evaluating the classification performance, i.e. cancer subtyping, cancer staging and gene mutation prediction tasks. The concordance index (C-Index) is calculated to evaluate the performance of the survival prediction task. The **bold** font and underlined font in all tables in the article represent the best and second-best performance, respectively.

C. Results and Analysis

We evaluate the effectiveness of the proposed SlideGCD by comparing its improvement on various state-of-the-art WSI analysis approaches and different downstream tasks. For the cancer subtyping and the cancer staging tasks, we selected 8 representative previous SOTA methods as baselines: *i)* ***ABMIL*** [10]: speaks for the classical lightweight attention-based MIL methods without considering patch relationships. *ii)* ***CLAM*** *[1]*: is the another influential classical attention-based MIL method that designs clustering-constrained-attention subnetwork for each category. *iii)* ***PatchGCN*** [13]: is a typical graph-based MIL method that involves the patch correlation via the patch-based graph. *iv)* ***TransMIL*** [16]: is a powerful transformer-based MIL method that exploits the patch correlations by utilizing the self-attention mechanism. *v)* ***DTFDMIL*** [2]: is a novel pseudo-bags-based MIL method that derives the instance probabilities as the attention score of each instance. *vi)* ***HiGT*** [4]: is a newly proposed multi-scale method that makes further consideration in hierarchical interaction across different magnifications via self-attention variant. *vii)* ***S4MIL*** [17]: is a novel sequence-based MIL method that firstly uses the structured state space models. *viii)* ***MaskHGL*** *[9]*: is a novel WSI classification framework that combines mask reconstruction strategy and hypergraph learning to achieve SOTA performance on the gene mutation prediction task with only H&E stained WSIs. For the survival prediction tasks, we selected the 4 influential benchmark methods, ***ABMIL***, ***PatchGCN***, ***TransMIL***, ***DTFDMIL*** as the baselines, and trained them with the negative log-likelihood (NLL) Loss following the setting of [13]. For the gene mutation prediction, we added the ***MaskHGL*** since it was originally proposed for the task of fine-grained gene mutation prediction. The experimental results for each downstream task are presented respectively in Tables I-IV.

Overall, the proposed SlideGCD is capable of substantially improving the accuracy of baseline models in most application scenarios. Especially for the pseudo-bag-based method DTFD-MIL, the SlideGCD significantly improved its performance on all four different downstream tasks, including but not limited to achieving the best performance on TCGA-BRCA (Subtyping) with 91.50% ACC, 92.83% AUC and 86.52% F1, and on TCGA-UCEC (Survival) with 67.53% C-Index. Moreover, it achieves the over 4% improvement of ACC & F1 on TCGA-NSCLC (Subtyping), and 9.77% of AUC gain on TCGA-

TABLE II: Comparisons on cancer staging. The ACCs are not reported because of the serious class imbalance in the cancer staging task, which makes the ACC of all methods approach the proportion of the class with more samples in the test set, thereby making the accuracy less referenceable.

Method	SlideGCD	TCGA-BRCA AUC(%)	TCGA-BRCA F1(%)	TCGA-NSCLC AUC(%)	TCGA-NSCLC F1(%)
ABMIL [10]	✗	58.75±1.84	46.00±3.15	63.98±1.96	58.24±3.56
	✔	60.79±3.18	48.24±6.36	67.57±2.02	58.03±6.48
	Δ	+2.04	+2.24	+3.59	-0.21
CLAM [1]	✗	57.96±2.70	50.47±5.04	62.18±1.88	56.86±4.33
	✔	60.01±2.91	53.00±2.51	65.48±4.39	58.79±4.33
	Δ	+2.05	+2.53	+3.30	+1.93
PatchGCN [13]	✗	57.96±1.80	49.28±4.61	59.66±1.63	53.46±5.66
	✔	**61.17±3.38**	51.54±1.99	65.85±5.17	53.36±5.89
	Δ	+3.21	+2.26	+6.19	-0.10
TransMIL [16]	✗	58.25±3.11	47.38±4.26	61.91±1.51	54.64±3.34
	✔	59.79±3.35	51.93±3.70	63.47±2.65	56.97±4.02
	Δ	+1.54	+4.55	+1.56	+2.33
DTFDMIL [2]	✗	58.14±2.52	53.72±1.37	59.73±1.74	54.74±2.72
	✔	60.23±1.78	54.25±2.86	**69.50±3.48**	58.64±6.23
	Δ	+2.09	+0.53	+9.77	+3.90
HiGT [4]	✗	56.23±1.58	50.86±4.44	58.77±2.65	50.65±2.25
	✔	56.41±2.85	49.72±5.63	59.44±3.60	50.09±5.78
	Δ	+0.18	-1.14	+0.67	-0.56
S4MIL [17]	✗	58.55±5.04	52.37±2.86	62.29±3.25	54.65±5.28
	✔	61.13±2.20	**54.69±2.63**	67.54±2.35	**59.55±5.21**
	Δ	+2.58	+2.32	+5.25	+4.90
MaskHGL [9]	✗	53.43±2.93	44.31±1.09	57.19±3.08	50.98±3.33
	✔	54.99±2.07	44.67±2.60	59.18±4.43	50.08±5.96
	Δ	+1.56	+0.36	+1.99	-0.90

NSCLC (Staging).

Each dataset we applied has its characteristics. For example, the dataset (TCGA-NSCLC) used in cancer subtyping corresponds to the most common binary classification without class imbalance. The cancer staging reveals the imbalanced binary classification and the fine-grained gene mutation prediction dataset (USTC-EGFR) represents the multi-class classification problem. Moreover, the regression problem is validated by the survival prediction task. All these multi-task experiments have proven the robustness and generalization of the proposed SlideGCD.

One exception is HiGT, a multi-scale method that further considers hierarchical interactions across different magnifications through a self-attention variant, in the cancer grading task. A foreseeable reason is the implicit contextual information at low magnification (e.g. thumbnails, 5×, and 10×) makes it difficult to achieve accurate cancer grading. Additionally, the unaligned patch clustering in HiGT, which is performed for each WSI separately, disturbs the connection measurement between WSIs and increases the heterogeneity of the slide-based graph. One another unideal situation occurred with MaskHGL where the improvements were not significant enough as well. The problem may be with the Masked Hypergraph ReConstruction (MHRC) module that is responsible for the mask reconstruction in MaskHGL. The learnable mask token in MHRC bridges the different slide representations since it participates in every round of training that involves the mask operation, thus the inter-correlation may have been potentially acquired to some extent in MaskHGL.

We also noticed that the improvement in the TCGA-BRCA cohort is smaller compared with TCGA-NSCLC in both cancer

TABLE III: Comparisons on survival prediction. All methods were trained with the NLL Loss following the setting of [13].

Method	SlideGCD	TCGA-BLCA C-Index	TCGA-UCEC C-Index
ABMIL [10]	✗	52.72±4.55	58.47±9.55
	✔	54.29±6.52	59.79±4.97
	Δ	+1.57	+1.32
PatchGCN [13]	✗	55.63±3.91	62.51±7.67
	✔	**57.45±4.00**	64.88±4.57
	Δ	+1.82	+2.37
TransMIL [16]	✗	54.50±1.90	60.21±9.12
	✔	55.24±2.34	65.64±7.24
	Δ	+0.74	+5.43
DTFDMIL [2]	✗	55.69±3.65	55.82±2.39
	✔	56.16±3.22	**67.53±5.35**
	Δ	+0.47	+11.71

subtyping and staging settings. A similar situation also occurred in two datasets for survival prediction: the improvement on the TCGA-BLCA dataset is smaller than that on the TCGA-UCEC dataset. Considering the varying difficulty levels of each dataset (e.g. the TCGA-BRCA&BLCA datasets appear more complex than TCGA-NSCLC&UCEC), it might suggest that the benefit brought by SlideGCD declines as the difficulty increases. The deeper reason is the separability of node embeddings is affected by the dataset difficulty and that narrows the potential of slide-based graph learning guided by the connection reflecting similar pathological patterns. However, even so, our SlideGCD can still achieve universal improvement in more complex downstream tasks, e.g. survival prediction and fine-grained gene mutation prediction.

D. Ablation Studies

In this section, we evaluate the effectiveness of the key components in our proposed SlideGCD, including the MIL head, graph branch, distillation, and CANB. Ablation experiments were conducted on the TCGA-BRCA dataset for subtyping and their results are shown in Table V. Due to the overall competitive performance, we chose DTFDMIL as the backbone for our ablation studies. We implemented three ablation variants for our method, each missing one or several key components of the SlideGCD. 1) The simplest variant (Line 2.) replaces the original classification head (MIL head) with the graph branch containing a dumb node buffer, which is updated via the FIFO strategy throughout training. Other variants sequentially add or replace new components on its basis. 2) The variant that is additionally equipped with "MIL Head" (Line 3.) adds the original slide-level classification head to accept the supervision from the inner-contextual perspective. 3) The variant with "MIL Head", "Graph Branch" and "Distillation" (Line 4.) is further armed with the response-based knowledge distillation loss to transfer the learned patch-level correlations into the graph branch. The last row indicates the full version of the SlideGCD (Line 5.) that replaces the dumb node buffer with the CANB and the update loss mentioned earlier is used to assist in buffer update.

From Table V, it can be seen that the performance of the proposed method is getting better with the addition of

TABLE IV: Comparisons on gene mutation prediction. Note that the public dataset TCGA-EGFR is class-imbalanced caused by the actual mutation rate.

Method	SlideGCD	USTC-EGFR ACC(%)	USTC-EGFR AUC(%)	USTC-EGFR F1(%)	TCGA-EGFR ACC(%)	TCGA-EGFR AUC(%)	TCGA-EGFR F1(%)
ABMIL [10]	✗	54.98±2.19	84.87±0.71	53.21±3.12	84.19±0.65	70.77±3.71	49.48±3.68
	✔	54.71±2.72	86.26±1.39	53.20±3.41	83.11±0.64	73.60±3.17	50.24±2.89
	Δ	-0.27	+1.39	-0.01	-1.08	+2.83	+0.76
PatchGCN [13]	✗	55.43±4.92	83.57±1.47	53.75±6.11	84.66±1.39	71.94±8.87	58.88±8.92
	✔	56.95±3.39	85.22±1.90	54.25±4.17	84.76±1.09	**79.30±3.98**	54.37±4.69
	Δ	+1.52	+1.65	+0.50	+0.10	+7.36	-4.51
TransMIL [16]	✗	55.61±3.82	85.73±1.96	53.14±5.38	84.73±1.97	73.72±6.72	58.23±7.14
	✔	60.81±2.90	86.14±1.40	60.20±3.01	**86.86±2.53**	76.15±2.82	**60.00±7.56**
	Δ	+5.20	+0.41	+7.06	+2.13	+2.43	+1.77
DTFDMIL [2]	✗	57.49±2.13	85.47±0.32	55.80±1.52	80.48±3.24	60.43±7.36	52.18±6.65
	✔	59.91±2.80	85.73±1.43	59.53±2.91	85.22±2.11	68.64±9.32	56.12±7.96
	Δ	+2.42	+0.26	+3.73	+4.74	+8.21	+3.94
MaskHGL [9]	✗	61.52±2.86	86.82±0.99	60.22±3.58	81.64±1.62	58.45±6.66	49.39±4.33
	✔	**62.24±3.20**	**87.51±1.31**	**61.56±3.76**	80.39±3.34	66.37±1.61	55.00±5.13
	Δ	+0.72	+0.69	+1.34	-1.25	+7.92	+5.61

TABLE V: Ablation on key components of the proposed SlideGCD. The reported results for SlideGCD variants all came from the predictions of its graph branch. The "CANB" denotes the Class-aware Node Buffer and its corresponding update strategy.

Methods	SlideGCD's Key Components: MIL Head	Graph Branch	Distillation	CANB	TCGA-BRCA (Subtyping) ACC(%)	AUC(%)	F1(%)
DTFDMIL (Baseline)	✓				89.30±0.44	90.08±0.86	83.17±1.43
SlideGCD (Ours)		✓			88.64±2.15	85.61±4.07	82.11±3.22
	✓	✓			89.97±1.08	91.97±0.76	84.62±2.06
	✓	✓	✓		90.43±1.65	91.23±1.26	85.19±2.59
	✓	✓	✓	✓	**91.50±0.50**	**92.83±0.93**	**86.52±0.78**

the proposed components. Specifically, comparing Lines 1-3., we can conclude that roughly replacing the MIL head with the graph branch cannot bring effective improvement, and the utilization of patch correlations is still crucial in our framework. It can be found in Lines 3&4 that knowledge distillation from the MIL branch can effectively increase the ACC (+0.46%) and F1-score (+0.57%) without involving new parameters. Last but not least, with the novel CANB and the designed buffer update strategy, the potential of the proposed framework has been further unleashed, as evidenced by a promising improvement of over 1% in all metrics in Lines 4&5. of Table V. In addition, the standard deviation of the results in cross-validation has also significantly decreased, we believe this is because the new buffer and buffer update strategy provide a more compact and reliable hidden space for adaptive graphing, ultimately resulting in more stable performance.

E. Model Verification

In this section, we discuss the effect of two ambiguous designs in SlideGCD and its generalization capacity against different patch encoders and application scenarios. *1) Why use distillation instead of fusion strategies? 2) Is the hypergraph representation necessary? Can we use the simple graph instead? 3) Does the improvement depend on a specific feature encoder? 4) How does the method work in scenarios where each patient contains multiple slides?*

TABLE VI: Ablation on multi-branch knowledge fusion strategy. We evaluated two distillation schemes and three common fusion schemes for maximizing the utilization of the dual-branch structure.

Interaction Strategy	TCGA-BRCA (Subtyping) ACC(%)	AUC(%)	F1(%)
DTFDMIL	89.30±0.44	90.08±0.86	83.17±1.43
SlideGCD-DTFDMIL			
w LogitsAddFusion	89.57±2.79	90.57±2.45	83.01±5.03
w FeatCatFusion	87.31±0.71	86.80±1.94	79.25±1.71
w FeatAddFusion	88.90±0.88	90.17±0.92	82.12±1.16
w Distillation (KLDiv)	**91.89±1.49**	**93.35±1.31**	**87.20±2.17**
w Distillation (JSDiv)	91.50±0.50	92.83±0.93	86.52±0.78

1) Distillation Vs. Fusion: There are many ways to gather knowledge from multiple branches of a neural network. The first idea is the Fusion strategy including Feature-level Fusion and Logits-level Fusion. In our case, the fusion strategy could be a substitute as long as it is well-performed and does not introduce new parameters. Therefore, comparisons were made between five classic strategies in Table VI including both fusion and distillation: *i)* ***LogitsAddFusion***: The outputted logits from both branches are added together for final predictions. *ii)* ***FeatCatFusion***: The embeddings outputted from the MIL backbone and the final graph convolutional layer are concatenated and then inputted to a linear layer to generate final logits for predictions. *iii)* ***FeatAddFusion***: The same

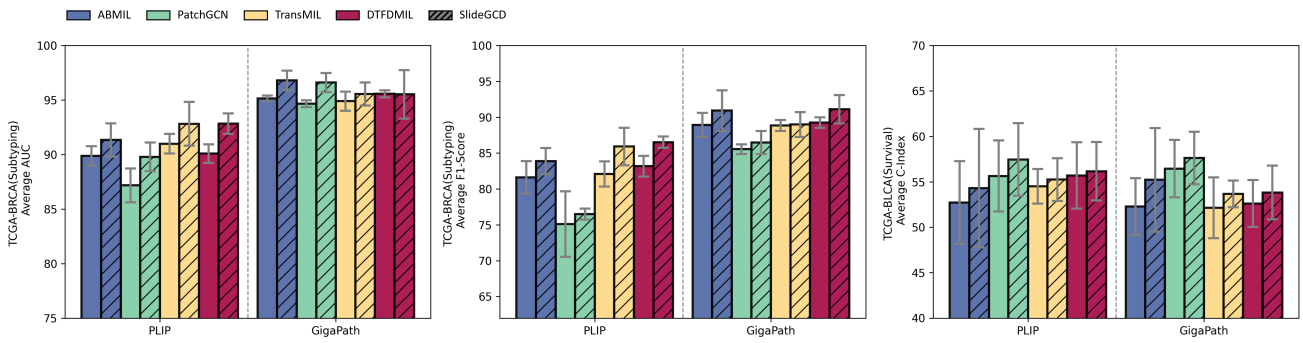

Fig. 4: Performance bar charts of different methods with different patch encoders on TCGA-BRCA (Subtyping) and TCGA-BLCA (Survival). We evaluated the performance of SlideGCD on different patch encoders (PLIP and GigaPath) with four reliable baselines. The bar charts and error lines represent the mean and standard deviation of metrics, respectively.

embeddings are sent to a project layer to align channels, and then added together to input a linear layer to generate final logits for predictions. *iv)* ***Distillation (KLDiv)***: The L_{JS} is replaced by the KL divergence loss following [48]. *v)* ***Distillation (JSDiv)***: The full version of SlideGCD. As shown in Table VI, only the LogitsAddFusion strategy slightly improved DTFDMIL with 89.57% ACC and 90.57% AUC among these three fusion strategies. Yet both common distillation strategies made significant performance rises for DTFDMIL without bringing extra computations. This demonstrates that distillation is more suitable for this occasion since the two branches are sensibly concerned with different aspects of information. Besides, the additional symmetry of JS divergence compared to KL divergence makes the network more stable and robust [45], which is reflected in the smaller standard deviations in Table VI. Considering the comparable performance of the two distillation strategies and the significantly smaller standard deviations of JSDiv implementation, we choose JS divergence loss as the default knowledge distillation setting of the proposed method.

2) Hypergraph Vs. Simple Graph: The difference between a hypergraph and a simple graph is their definition of edge where the hyperedge can connect more than two nodes. From this perspective, the simple graph can be viewed as a special case of hypergraph. Then could the SlideGCD pipeline take effect without the hypergraph representation and the hypergraph convolutional layers? To answer this question, we conducted comparisons that replaced the hypergraph and related layers with the simple graph and three widely used graph convolution layers. As shown in Table VII, it is gratifying that all modifications of the SlideGCD work properly and bring visible performance increase to the baseline. In conclusion, we have demonstrated the improvements brought by SlideGCD are not bound to any specific graph convolution operation but are achieved by the framework itself.

3) Generalization against different features: To further explore the generalizability of SlideGCD, we additionally applied it to three more approaches that comprise feature generation based on different training paradigms: *i)* ***HIPT*** *[55], a vision transformer that utilizes hierarchical self-supervised learning to generate patch embeddings; ii)* ***PANTHER*** *[56], a novel method that could build the task-agnostic slide representations in a completely unsupervised fashion; iii)* ***GigaPath*** *[57], a novel whole-slide pathology foundation model pretrained on 1.3 billion 256×256 pathology image tiles in 171,189 whole slides*. Regarding experiment settings, we chose the TCGA-BRCA and the TCGA-BLCA datasets to evaluate their performance in both cancer subtyping and survival prediction tasks. For the foundation model GigaPath, we used it as the patch encoder and applied four reliable benchmark methods on its base. For HIPT and PANTHER, we retrained these methods on the target datasets and considered their decision-making model as the MIL branch of the SlideGCD to assess its performance. All experiments follow the batch size of 64 and their original parameter setting. The results are reported in Table VIII and Fig.4

TABLE VII: Ablation on different graph convolution operations. We switched the framework back to the simple graph style and evaluated three popular graph convolution layers to prove that the consistent improvements brought by SlideGCD are not bound to any specific graph convolution operation.

GraphConv Operation	TCGA-BRCA (Subtyping) ACC(%)	AUC(%)	F1(%)
DTFDMIL	89.30±0.44	90.08±0.86	83.17±1.43
SlideGCD-DTFDMIL			
w GCNConv [49]	90.76±0.74	91.27±1.25	84.74±1.13
w GATConv [50]	90.17±0.77	91.57±1.94	83.74±1.90
w GINConv [51]	89.83±2.67	91.03±2.13	84.42±3.75
w HyperGraphConv [42]	**91.50±0.50**	**92.83±0.93**	**86.52±0.78**

The experiment results demonstrate that the SlideGCD could provide stable improvements on most occasions and also prove the improvement does not rely on specific feature encoders. The proposed method is not only applicable to the features provided by the foundation models trained on huge amounts of data, but also to the features provided by self-supervised training methods which are trained on a relatively small dataset and can even be directly applied to the unsupervised task-agnostic slide-level representations.

4) Generalization against multiple slides: In clinical practice, the final diagnosis is issued on a case-by-case basis. A

TABLE VIII: Performance of SlideGCD on hierarchical self-supervised learning method – HIPT and unsupervised slide representation learning method – PANTHER. We retrained these methods on the target datasets (TCGA-BRCA & TCGA-BLCA) and took their decision-making module as the MIL branch of the SlideGCD.

Method	SlideGCD	TCGA-BRCA (Subtyping) AUC(%)	TCGA-BRCA (Subtyping) F1(%)	TCGA-BLCA (Survival) C-Index
HIPT [55]	✗	79.85±3.72	69.62±5.62	53.82±6.00
	✔	80.32±1.98	72.15±1.95	54.55±4.71
	Δ	+0.47	+2.53	+0.73
PANTHER [56]	✗	88.94±1.10	81.83±1.79	53.59±1.73
	✔	90.53±1.38	83.87±2.23	57.03±2.94
	Δ	+1.59	+2.04	+3.44

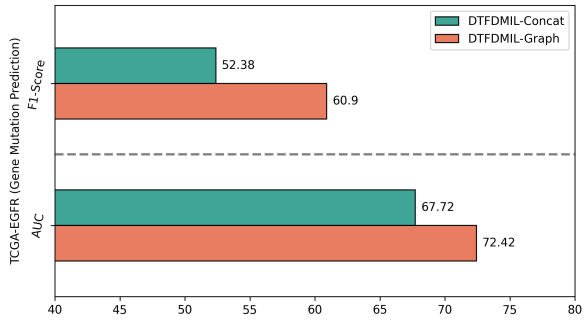

Fig. 5: Performance on multiple-slide patient-level prediction task. We evaluated the proposed method based on DTFDMIL in the TCGA-EGFR patient-level prediction task where each patient contains more than one slide.

normal situation is that a patient possesses multiple WSIs. A reasonable solution needs to consider all WSIs for a finer decision. Since the performance of the slide-based graph in slide-level prediction tasks has been verified, it is interesting to see how the method works in scenarios where each patient contains multiple slides. Therefore, we designed a simple experiment to verify the generability of our method against multiple slides in the patient-level diagnosis prediction task. We evaluated two kinds of patient representation generation strategies, differentiating whether the slide-based graph was involved, on the TCGA-EGFR gene mutation prediction dataset (in which each patient contains multiple WSIs) with the modified DTFDMIL and DTFDMIL-SlideGCD. For the baseline DTFDMIL, denoted as DTFDMIL-Concat in Fig.5, we directly input the patch embeddings of each slide belonging to the same patient in parallel, after the slide embeddings are produced, an additional gated-attention module is adopted to aggregate the slide embeddings to the patient-level representation, the original classifier is involved in making the patient-level prediction in the end. As to the DTFDMIL-SlideGCD, denoted as DTFDMIL-Graph in Fig.5, we followed the same strategy described above. The difference is all of the generated slide embeddings will be inserted into the slide-based graph and the message-passing will be conducted in the slide-based graph to refine the slide embeddings. After that, the same gated-attention module is also involved for the same purpose.

From Fig.5, it can be noticed that the performance of the DTFDMIL-SlideGCD surpasses the baseline by 4.7% of AUC and 8.52% of F1-score, which proves the slide-level message-passing could also benefit the aggregation of patient-level representations and showing the generalization ability of the proposed methods in the practice of multiple slides situation.

F. Hyper-parameters Analysis

In this section, we systematically explore the effect of hyper-parameters in SlideGCD with the backbone of DTFDMIL on the TCGA-BRCA (Subtyping) task. The results come from the five-fold cross-validation on the validation set. The impact of the following hyper-parameters will be discussed: 1) the size of node buffer L, 2) the nearest neighbors k, 3) the distillation temperature t, 4) the loss weight of buffer update β, and 5) the temperature in buffer update τ.

1) Size of node buffer L: L determines the exact number of nodes in the slide-based graph and greatly affects the capacity of the class-aware sub-queue. Since the success of graph neural networks is believed to be rooted in the homophily assumption [52]–[54], we need to ensure that the slide-based graph has a certain degree of homogeneity. Thus we allow the buffer to be larger than the capacity of the training set, so that there can be more nodes in the graph that actually represent the same WSI thereby increasing homogeneity. However, when L is set too large, the node buffer will contain much more outdated slide embeddings that might defect the performance of the SlideGCD and will introduce more extra computation as well. When it is too small, the effect of the slide-based graph might be inapparent since there is a lack of sufficient homogeneous information in the slide-based graph. We tuned L in the range of [1024, 4096] with a step of 512. The curve in Fig. 6(A) shows that the performance of SlideGCD is indeed altered, but it can still stably surpass the baseline. Eventually, we choose $L = 3072$ as the optimal value for balancing the accuracy and computational complexity.

2) Nearest neighbors k: k is another important hyper-parameter that controls the connection density in the slide-based graph and thus influences the message passing during graph learning. Experimentally, we tuned $k \in [6, 8, 10, 12, 14, 16]$ for verification. From Fig. 6(B), it is known that SlideGCD is less sensitive to k than L since the validated AUCs are quite steady and always at a relatively high level compared with the baseline. We choose $k = 12$ as the default setting for a better AUC.

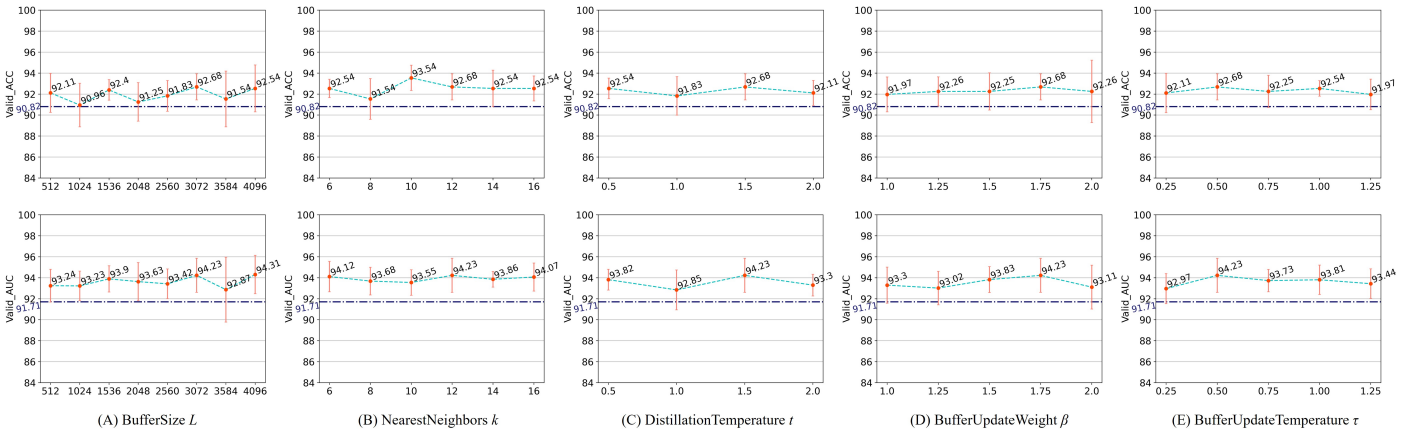

Fig. 6: Performance curves on the validation set in the five-fold cross-validation, where the error lines indicate the standard deviation of the metrics and the dashed lines parallel to the horizontal axis represent the metrics that the baseline (DTFDMIL) can achieve.

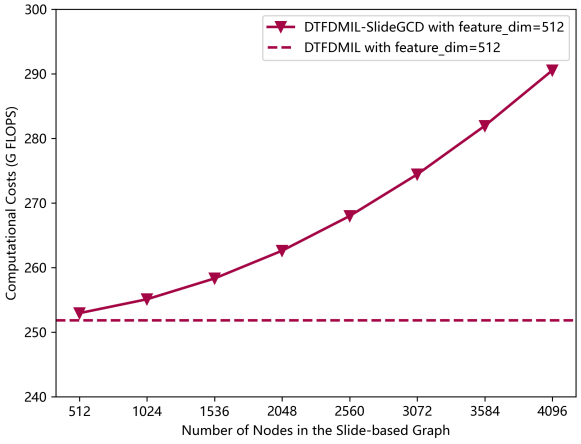

Fig. 7: Quantitative comparison of computational costs on different settings of node number (buffer size).

3) Distillation temperature t: This hyper-parameter manages the effect of distillation. In a higher temperature situation (i.e. $t > 1$), distillation focuses on transferring knowledge from the teacher model. When getting a lower temperature (i.e. $t < 1$), distillation tends to alleviate the impact of noise in negative samples [48]. In our application, our objective is to transfer the well-learned knowledge in MIL head to the SlideGNN, thus a relatively large temperature coefficient should have a better effect. Following the results in Fig. 6(C), we set $t = 1.5$.

4) Loss weight of buffer update β: β controls the contribution from the buffer update. The balance between each component of the final loss is important for achieving the best performance. We tested it in the range of [1.0, 2.0] with a step of 0.25 since the buffer update is not the main supervision signal in our framework. As shown in Fig. 6(D), the model performance is not sensitive to it. Empirically, we set $\beta = 1.75$.

5) Temperature in buffer update loss τ: τ supervises the effect of contrastive learning. In SlideGCD, the node embeddings continuously shift during training resulting in many noises and ultimately leading to heterogeneity in the slide-based graph. Under the circumstances, we wish the buffer update strategy which is based on contrastive learning could alleviate the impact of noise brought by the over-shifting slide embeddings. The curves in Fig. 6(E) validate the rationality of the motivation since the performances reach the peak with a lower temperature $\tau = 0.5$.

G. Computational Complexity

Since the graph branch in the proposed SlideGCD is completely additional when compared with the baseline, it is inevitable that our framework requires more computational resources. The overall additional computational complexity of the proposed framework is $O((Nd^2 + N^2d) + (N^2 + NMd + NKd + Nd^2))$, where N is the number of nodes(slides) in the slide-based graph, d is the dimension of the slide-level embeddings, M is the number of hyperedges in the slide-based graph, K is the average neighborhood size. Specifically, the time complexity of the Adaptive Graph Construction is $O(Nd^2 + N^2d)$, which consists of linear projections and the k-NN clustering. The graph encoder SlideGNN takes $O(N^2 + NMd + NKd + Nd^2)$, when the hypergraph convolution with hypergraph attention [42] is adopted. As the slide-based graph is constructed by k-NN strategy, the number of hyperedges M equals the number of nodes N, and the average neighborhood size K equals the hyper-parameter k ($K \ll N$). Then, the simplified computational complexity is $O(N^2d + Nd^2)$. The buffer update and its loss calculation take $O(N)$. Our framework does not involve noticeable memory cost during training since the addition parameters only come from a small number of linear layers and graph convolutional layers and the node buffer does not associate with gradient-backward.

In Fig.7, We have summarized the changing trend of network computation with the number of nodes in the slide-based graph. We have analyzed the computational costs of

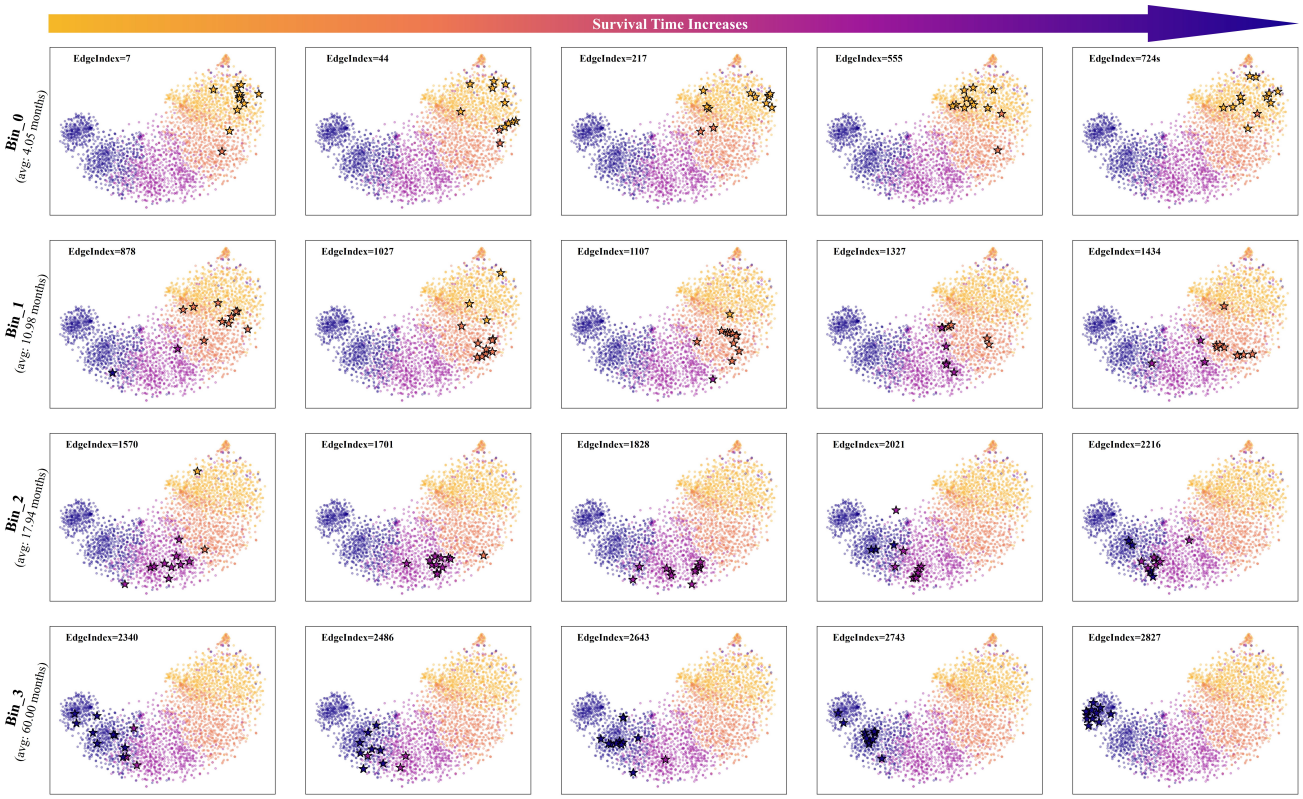

Fig. 8: T-SNE visualization for the node buffer from TCGA-BLCA survival prediction task. Each point corresponds to a patient and its color goes deeper as the survival time increases. The stars in each sub-figure are associated with one hyperedge of the slide-based graph. For the 4 bins setting, we present its average survival time and highlight 5 hyperedges for each bin.

the SlideGCD under different node buffer sizes. The reported results are from a simulated mini-batch that consists of 64 WSIs with 5000 patches. It can be observed that the additional computational cost of the network shows a non-linear growth trend with the number of nodes in the slide-based graph. Considering the hyper-parameters analysis of buffer size L in Fig.6(A), we generally don't need a particularly large buffer size (e.g. exceeds 4096) to achieve a better performance than the baseline. Therefore, most of the time the actual additional computational cost of SlideGCD will be acceptable.

H. Interpretability and Visualization

To intuitively discuss the effect of the slide-based graph learning, we visualize the overall distribution of the node embeddings within the node buffer (on TCGA-BLCA) via T-SNE where we highlight 20 hyperedges to assess its node distribution. Following the discretization of survival time in [13], the patients are normally divided into 4 buckets (bins: 0-3) as the increment of survival time. As shown in Fig. 8, all sub-figures share the same background which represents the overall distribution of the node embeddings in the buffer where each point indicates a node, and the color goes deeper as the survival time extends. The pointed stars are from the same hyperedge and the index is above each subgraph.

From the overall background of Fig. 8, it can be clearly observed that the class-aware node buffer can significantly separate the nodes from different buckets and the distribution of the node embeddings follows a pattern that the survival time of nodes gradually extends in a circular arc shape from top right to bottom and then to left. Inspecting each sub-figure, nodes are more inclined to associate with other nodes in the same bucket (having close survival times). As the survival time increases (the edge index grows), the center of the hyperedge is shifting along that pattern. The above observations intuitively validate that the proposed pipeline has achieved the association of homogeneous samples and also prove the improvement brought by SlideGCD is explainable and traceable.

V. Conclusion

In this paper, we present an end-to-end generic pipeline SlideGCD for histopathology whole slide image analysis, which exploringly takes the unrestricted slide inter-correlations into account via the slide-based graph and proves its consistent improvements. The rehearsal-based adaptive graph construction strategy we devised, models the WSI dataset into a slide-based graph where WSIs are nodes and their connections can be updated during training. Besides, knowledge distillation is applied to train MIL and the graph branch collaboratively and try not to lose the inner-contextual knowledge learned by Head_{MIL} as much as possible. Extensive experiments and visualizations are conducted to demonstrate the effectiveness and robustness in an interpretable way.

Although the proposed SlideGCD achieved promising improvements on various backbones and downstream tasks, some

upgrades still could be made. For example, 1) buffer update strategies based on other principles, such as uncertainty, may optimize the training and inference overhead; 2) the further exploration of slide-level message-passing for patient-level prediction tasks has its worth. We hope that this work can attract much attention to the exploration of the slide-level interaction and we believe this will assist in the advancement of foundation models for computational pathology.

REFERENCES

[1] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nat Biomed Eng.*, vol. 5, no. 6, pp. 555–570, 2021.

[2] H. Zhang, Y. Meng, Y. Zhao, Y. Qiao, X. Yang, S. E. Coupland, and Y. Zheng, "Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18802–18812.

[3] G. Bontempo, F. Bolelli, A. Porrello, S. Calderara, and E. Ficarra, "A graph-based multi-scale approach with knowledge distillation for wsi classification," *IEEE Trans. Med. Imag.*, vol. 43, no. 4, pp. 1412-1421, April 2024.

[4] Z. Guo, W. Zhao, S. Wang, and L. Yu, "Higt: Hierarchical interaction graph-transformer for whole slide image analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2023, pp. 755–764.

[5] J. Shi, L. Tang, Y. Li, X. Zhang, Z. Gao, Y. Zheng, C. Wang, T. Gong, and C. Li, "A structure-aware hierarchical graph-based multiple instance learning framework for pt staging in histopathological image," *IEEE Trans. Med. Imag.*, vol. 42, no. 10, pp. 3000-3011, Oct. 2023.

[6] L. Fan, A. Sowmya, E. Meijering, and Y. Song, "Cancer survival prediction from whole slide images with self-supervised learning and slide consistency," *IEEE Trans. Med. Imag.*, vol. 42, no. 5, pp. 1401-1412, May 2023.

[7] W. Hou, C. Lin, L. Yu, J. Qin, R. Yu, and L. Wang, "Hybrid graph convolutional network with online masked autoencoder for robust multimodal cancer survival prediction," *IEEE Trans. Med. Imag.*, vol. 42, no. 8, pp. 2462-2473, Aug. 2023.

[8] Y. Zheng, K. Wu, J. Li, K. Tang, J. Shi, H. Wu, Z. Jiang, and W. Wang, "Partial-label contrastive representation learning for fine-grained biomarkers prediction from histopathology whole slide images," *IEEE J. Biomed. Health Inform.*, 2024.

[9] J. Shi, T. Shu, K. Wu, Z. Jiang, L. Zheng, W. Wang, H. Wu, and Y. Zheng, "Masked hypergraph learning for weakly supervised histopathology whole slide image classification," *Comput Methods Programs Biomed.*, vol. 253, p. 108237, 2024.

[10] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.

[11] P. Liu, L. Ji, X. Zhang, and F. Ye, "Pseudo-bag mixup augmentation for multiple instance learning-based whole slide image classification," *IEEE Trans. Med. Imag.*, vol. 43, no. 5, pp. 1841-1852, May 2024.

[12] R. Li, J. Yao, X. Zhu, Y. Li, and J. Huang, "Graph cnn for survival analysis on whole slide pathological images," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2018, pp. 174–182.

[13] R. J. Chen, M. Y. Lu, M. Shaban, C. Chen, T. Y. Chen, D. F. Williamson, and F. Mahmood, "Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2021, pp. 339–349.

[14] Y. Guan, J. Zhang, K. Tian, S. Yang, P. Dong, J. Xiang, W. Yang, J. Huang, Y. Zhang, and X. Han, "Node-aligned graph convolutional network for whole-slide image representation and classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 18813–18823.

[15] W. Lu, M. Toss, M. Dawood, E. Rakha, N. Rajpoot, and F. Minhas, "Slidegraph+: Whole slide image level graphs to predict her2 status in breast cancer," *Med. Image Anal.*, vol. 80, p. 102486, 2022.

[16] Z. Shao, H. Bian, Y. Chen, Y. Wang, J. Zhang, X. Ji, *et al.*, "Transmil: Transformer based correlated multiple instance learning for whole slide image classification," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2136–2147.

[17] L. Fillioux, J. Boyd, M. Vakalopoulou, P.-H. Cournède, and S. Christodoulidis, "Structured state space models for multiple instance learning in digital pathology," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2023, pp. 594–604.

[18] S. Yang, Y. Wang, and H. Chen, "Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology," 2024, *arXiv:2403.06800*.

[19] T. H. Chan, F. J. Cendra, L. Ma, G. Yin, and L. Yu, "Histopathology whole slide image analysis with heterogeneous graph representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 15661–15670.

[20] C. Rivera and B. Venegas, "Histological and molecular aspects of oral squamous cell carcinoma," *Oncol Lett.*, vol. 8, no. 1, pp. 7–11, 2014.

[21] K. Simon, "Colorectal cancer development and advances in screening," *Clin Interv Aging.*, vol. 11, pp. 967–976, 2016.

[22] Z. Seferbekova, A. Lomakin, L. R. Yates, and M. Gerstung, "Spatial biology of cancer evolution," *Nat Rev Genet.*, vol. 24, no. 5, pp. 295–313, 2023.

[23] Z. Shao, Y. Chen, H. Bian, J. Zhang, G. Liu, and Y. Zhang, "Hvtsurv: Hierarchical vision transformer for patient-level survival prediction from whole slide image," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023, pp. 2209–2217.

[24] D. Reisenbüchler, S. J. Wagner, M. Boxberg, and T. Peng, "Local attention graph-based transformer for multi-target genetic alteration prediction," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2022, pp. 377–386.

[25] Y. Zheng, R. H. Gindra, E. J. Green, E. J. Burks, M. Betke, J. E. Beane, and V. B. Kolachalama, "A graph-transformer for whole slide image classification," *IEEE Trans. Med. Imag.*, vol. 41, no. 11, pp. 3003-3015, Nov. 2022.

[26] D. Di, J. Zhang, F. Lei, Q. Tian, and Y. Gao, "Big-hypergraph factorization neural network for survival prediction from whole slide image," *IEEE Trans. Image Process*, vol. 31, pp. 1149–1160, 2022.

[27] D. Di, C. Zou, Y. Feng, H. Zhou, R. Ji, Q. Dai, and Y. Gao, "Generating hypergraph-based high-order representations of whole-slide histopathological images for survival prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5800–5815, 2022.

[28] W. Hou, H. Huang, Q. Peng, R. Yu, L. Yu, and L. Wang, "Spatial-hierarchical graph neural network with dynamic structure learning for histological image classification," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2022, pp. 181–191.

[29] P. Liu, L. Ji, F. Ye, and B. Fu, "Graphlsurv: A scalable survival prediction network with adaptive and sparse structure learning for histopathological whole-slide images," *Comput Methods Programs Biomed.*, vol. 231, p. 107433, 2023.

[30] J. Li, Y. Chen, H. Chu, Q. Sun, T. Guan, A. Han, and Y. He, "Dynamic graph representation with knowledge-aware attention for histopathology whole slide image analysis," 2024, *arXiv:2403.07719*.

[31] F. Li, M. Wang, B. Huang, X. Duan, Z. Zhang, Z. Ye, and B. Huang, "Patients and slides are equal: A multi-level multi-instance learning framework for pathological image analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2023, pp. 63–71.

[32] W. Xinggang, Y. Yongluan, T. Peng, L. Wenyu, and G. Xiaojie, "Bag similarity network for deep multi-instance learning," in *Information Sciences*, vol. 504, pp. 578–588, 2019.

[33] P. Keller, M. Dawood, B. S. Chohan, and M. Fayyaz, "HistoKernel: Whole slide image level maximum mean discrepancy kernels for pan-cancer predictive modelling," in *Med Image Anal.*, vol. 101, p. 103491, 2025.

[34] Y. Rui, L. Pei, J. Luping, "ProDiv: Prototype-driven consistent pseudo-bag division for whole-slide image classification," in *Comput Methods Programs Biomed.*, vol. 249, p. 108161, 2024.

[35] J. Gou, L. Ji, P. Liu, and M.Ye, "Queryable prototype multiple instance learning with vision-language models for incremental whole slide image classification," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2025.

[36] Z. Huang, F. Bianchi, M. Yuksekgonul, T. J. Montine, and J. Zou, "A visual–language foundation model for pathology image analysis using medical twitter," *Nat Med.*, vol. 29, no. 9, pp. 2307–2316, 2023.

[37] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9729–9738, 2020.

[38] J. Li, Y. Zheng, K. Wu, J. Shi, F. Xie, and Z. Jiang, "Lesion-aware contrastive representation learning for histopathology whole slide images analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2022, pp. 273–282.

[39] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: defying

forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, 2021.

[40] Y. Huang, W. Zhao, S. Wang, Y. Fu, Y. Jiang, and L. Yu, "Conslide: Asynchronous hierarchical interaction transformer with breakup-reorganize rehearsal for continual whole slide image analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 21349–21360.

[41] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[42] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, p. 107637, 2021.

[43] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[44] J. Lin, "Divergence measures based on the shannon entropy," in *IEEE Trans. Inf. Theory*, vol.37, pp. 145–151, 1991.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets," in *Adv. Neural Inf. Process Syst.*, vol.27, 2014.

[46] C. Kandoth, M. D. McLellan, F. Vandin, K. Ye, B. Niu, C. Lu, *et al.*, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, 2013.

[47] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC Med. Res. Methodol.*, vol. 18, pp. 1–12, 2018.

[48] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[49] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[50] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *Proc. Int. Conf. Learn. Represent.*, 2018.

[51] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," 2018, *arXiv:1810.00826*.

[52] S. Luan, C. Hua, M. Xu, Q. Lu, J. Zhu, X. Chang, J. Fu, J. Leskovec, and D. Precup"When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability,", *Adv. Neural Inf. Process Syst.*, vol. 36, 2024.

[53] X. Li, R. Zhu, Y. Cheng, C. Shan, S. Luo, D. Li, and W. Qian, "Finding global homophily in graph neural networks when meeting heterophily," *International Conference on Machine Learning*, 2022.

[54] M. McPherson, L. Smith-Lovin, and J. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, pp. 415–444, 2001.

[55] R. Chen, C. Chen, Y. Li, T. Chen, A. Trister, R. Krishnan, F. Mahmood, "Scaling vision transformers to gigapixel images via hierarchical self-supervised learning," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 16144–16155, 2022.

[56] A. Song, R. Chen, T. Ding, D. Williamson, G. Jaume, F. Mahmood, "Morphological prototyping for unsupervised slide representation learning in computational pathology," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11566–11578, 2024.

[57] H. Xu, N. Usuyama, J. Bagga, S. Zhang, R. Rao, T. Naumann, C. Wong, Z. Gero, J. González, G. Javier, Y. Gu and others, "A whole-slide foundation model for digital pathology from real-world data," *Nature*, vol. 630, pp. 181–188, 2024.