

UCLA

UCLA Electronic Theses and Dissertations

Title

Graph Deep Learning for Pathology Imaging and Heterogeneous Medical Data

Permalink

<https://escholarship.org/uc/item/0hn010q6>

Author

Wang, Zichen

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Graph Deep Learning for Pathology Imaging and Heterogeneous Medical Data

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Bioengineering

by

Zichen Wang

2024

© Copyright by
Zichen Wang
2024

ABSTRACT OF THE DISSERTATION

Graph Deep Learning for Pathology Imaging and Heterogeneous Medical Data

by

Zichen Wang

Doctor of Philosophy in Bioengineering

University of California, Los Angeles, 2024

Professor Corey Wells Arnold, Chair

Graphs are a universal language for describing systems composed of interacting elements. In biomedicine, graph-structured data are pervasive, ranging from molecular and cellular interaction networks to healthcare knowledge graphs. Graph neural networks (GNNs) have initiated a new paradigm for graph representation learning. Moreover, the Transformer model, conceptually a specialized form of GNN, has been pushing the boundaries in the most impactful domains of machine learning over the past half-decade. These domains range from language modeling and computer vision to computational biology. However, the complex nature of biomedical data poses significant technical challenges for graph learning, including heterogeneity, multi-scale interactions, and limited data annotations. This dissertation attempts to address these obstacles across various biomedical applications, including pathology image analysis and heterogeneous data integration. We first introduce a bipartite graph convolutional network for drug repurposing through heterogeneous information fusion. Additionally, we present a temporal graph learning method that leverages heterogeneous electronic health record (EHR) to improve depression prediction. We further advance graph deep learning techniques for histopathology image analysis across multiple magnification levels,

including patient-level survival prediction and tissue-level tumor microenvironment modeling. Lastly, inspired by recent advances in vector-quantized image modeling and Generative Pretrained Transformer (GPT), we propose a self-supervised learning framework tailored for computational pathology. Our approach explores pretraining and fine-tuning strategies to improve generalizable pathology image representation learning, thus facilitating the development of data-efficient deep learning models for various clinical workflows. Together, these methods demonstrate the versatility and effectiveness of graph deep learning in addressing complex biomedical challenges, paving the way for real-world applications in disease modeling, diagnosis, and treatment optimization.

The dissertation of Zichen Wang is approved.

Aydogan Ozcan

Dan Ruan

Yingnian Wu

William F. Speier

Corey Wells Arnold, Committee Chair

University of California, Los Angeles

2024

Dedicated to my family and friends
Special gratitude to my beloved parents

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivation	3
1.1.1	Enhancing Graph Learning for Heterogeneous Information Fusion in Drug Repurposing and EHR Modeling	3
1.1.2	Improving Graph Learning for Multi-scale Pathology Image Analysis	4
1.1.3	Advancing Generalizable Image Representation Learning for Compu- tational Pathology	5
1.2	Contributions	6
1.3	Outline of the Dissertation	8
2	Background	10
2.1	Graph Neural Network	10
2.1.1	Inductive Bias	10
2.1.2	Computational Framework	12
2.1.3	Design Pipeline	15
2.2	Transformer	18
2.2.1	Original Model Architecture	18
2.2.2	Model Usage	22
2.2.3	Connections with Graph Neural Network	22
2.3	Self-supervised Learning	23
2.3.1	Contrastive Learning Approaches	24
2.3.2	Generative Learning Approaches	25

2.3.3	Masked Modeling	27
3	Bipartite Graph Convolutional Networks for In Silico Drug Repurposing	29
3.1	Abstract	29
3.2	Introduction	30
3.3	Related Work	33
3.4	Datasets	36
3.4.1	Drug-Disease Associations	37
3.4.2	Drug-Protein and Disease-Protein Associations	37
3.4.3	Protein-Protein Interactions	38
3.5	Method	38
3.5.1	Inter-Domain Message Passing through Bipartite Graph Convolution	38
3.5.2	Model Framework	39
3.5.3	Node Feature Representation	41
3.5.4	Bipartite Graph Convolutional Encoder	41
3.5.5	Multi-layer Perceptron Neural Network Decoder	43
3.5.6	Model Training	43
3.6	Experiments	44
3.6.1	Evaluation Metrics	44
3.6.2	Method Comparison	44
3.7	Results	46
3.7.1	Performance Comparison	46
3.7.2	Investigation of Novel Predictions	46
3.7.3	Experiments on the External Dataset	47

3.7.4	Case Study	48
3.7.5	The Effect of Layer Numbers on Model Performance	50
3.8	Conclusion	51
4	Temporal Graph Learning using Heterogeneous Electronic Health Records to Predict Depression	52
4.1	Abstract	52
4.2	Introduction	53
4.3	Related Work	54
4.3.1	Temporal Deep Learning for EHR	55
4.3.2	Temporal Graph Neural Network	55
4.4	Problem Formulation	57
4.4.1	Temporal Heterogeneous Graph	57
4.4.2	Graph Representation of EHR	57
4.5	Data Preprocessing	58
4.5.1	Sequential Dataset Construction	58
4.5.2	Graph Dataset Construction	60
4.6	Methods	61
4.6.1	Graph Node Embedding	61
4.6.2	Time-aware Information Propagation	62
4.6.3	Heterogeneous Graph Attention	63
4.6.4	Subgraph Sampling	65
4.6.5	Model Training	66
4.7	Experiments	67

4.7.1	Experimental Setup	67
4.7.2	Baseline Methods	67
4.8	Results	68
4.8.1	Performance Comparison	68
4.8.2	Graph Model Explainability	71
4.9	Discussion	71
4.10	Conclusion	73
5	Hierarchical Graph Pathomic Network for Progression Free Survival Prediction	74
5.1	Introduction	74
5.2	Method	76
5.2.1	Tumor Region Detection	76
5.2.2	Graph Pathomic Network	77
5.2.3	Model Training for Progression Free Survival Prediction	79
5.2.4	Momentum Contrast for Unsupervised Pretraining of Cell Graph	79
5.3	Experiment	81
5.3.1	Dataset and Preprocessing	81
5.3.2	Methods Comparison	81
5.3.3	Model Training and Evaluation	82
5.4	Results	83
6	Graph Deep Learning for Tumor-associated Stroma Identification in Prostate Histopathology Slides	85
6.1	Introduction	85

6.2	Method	87
6.2.1	Stroma Tissue Segmentation	87
6.2.2	Stroma Classification with Spatial Patch Graphs	88
6.2.3	Neighbor Consistency Regularization for Noisy Labels	89
6.2.4	Adversarial Multi-modal Learning	91
6.3	Experiment	92
6.3.1	Dataset	92
6.3.2	Model Training and Evaluation	94
6.4	Results and Discussions	94
7	Self-supervised Learning for Digital Pathology using Vector-Quantized Im-	
	age Modeling	96
7.1	Introduction	96
7.2	Methods	97
7.2.1	Whole Slide Image Representation	97
7.2.2	Learning Visual Tokenizer using VQGAN	99
7.2.3	Modeling Image Composition with Scale-aware Transformer	100
7.2.4	Unifying Image Generation and Multi-scale Representation Learning .	102
7.3	Experiment Setup	105
7.3.1	Model Pretraining	105
7.3.2	Finetune Strategies for Downstream Tasks	106
7.4	Experiment Results	107
7.4.1	Patch-level Image Classification	107
7.4.2	Patch-level Semantic Segmentation	111

7.4.3	Whole Slide Image Classification	114
7.5	Discussion	117
7.5.1	Histopathology Image Reconstruction and Synthesis	117
7.5.2	Impact of Transformer Layer on Representation Quality	119
7.6	Conclusion	120
8	Conclusion and Future Work	121
8.1	Summary of Contributions	121
8.2	Future Work	123
8.2.1	Systematic Evaluation of Graph Model Explainability	123
8.2.2	Graph Structure Learning and Dynamic Graph Modeling	123
8.2.3	Multi-modal Self-supervised Learning Framework	124
	References	125

LIST OF FIGURES

2.1	Overview of the original Transformer architecture, reproduced from [LWL22]. . .	19
3.1	Overview of our heterogeneous information network. The multi-relational network has 592 disease, 1,012 drug and 13,460 protein nodes connected by 3,204 drug-disease, 7,713 drug-protein, 104,716 disease-protein and 141,296 protein-protein edges.	36
3.2	Overview of BiFusion model architecture. (A) The pipeline of BiFusion contains a bipartite graph convolutional network encoder and a MLP decoder. The encoder takes similarity features of drug and disease nodes as inputs, and generates drug-disease pair embeddings by fusing heterogeneous information through message passing across drug, disease and protein nodes. Each BiFusion layer consists of three computing steps shown in the following sub-figures. BiFusion decoder takes pair embeddings to produce prediction score and reconstruct drug-disease association matrix. (B) The first step in BiFusion layer: a single bipartite graph attention convolution layer is applied to project information from drug and disease domains to protein domain. (C) The second step in BiFusion layer: a single layer graph attention convolution layer is applied within PPI network. (D) The third step in BiFusion layer: another bipartite graph attention convolution layer is used to update drug and disease features based on learnt protein node embeddings. .	40
3.3	Effect of the number of layers on model performance. The x axis denotes the number of BiFusion layers and the y axis is the model performance on testing set.	50
4.1	Illustration of transforming EHR sequences into temporal graphs. Our approach samples temporal subgraphs from patient-code bipartite graphs to more effectively capture high-order collaborative signals among patient sequences.	54

4.2	Overview of proposed model architecture. Temporal encoding module quantifies temporal effects using edge timestamps in a continuous space. Details of heterogeneous graph attention can be found in Fig. 4.3.	61
4.3	Illustration of heterogeneous graph attention mechanism.	64
4.4	Temporal analysis of edge attention weight distribution. A smaller timespan indicates a more recent code record.	72
5.1	Model architecture of the hierarchical graph pathomic network. Given a WSI, we sample patches using the tumor detection model to construct a patch graph, each node of which is represented by its corresponding cell graph. Our model learns hierarchical graph representations of WSIs from nuclei-level to patch-level. . . .	77
6.1	Process of stroma patch graph construction: The left prostate model illustrates the locations of biopsy and whole-mount tissues in 3D space. Biopsy slides provide a targeted view while whole-mount slides offer a broader perspective of the tumor and surrounding tissue. The stroma segmentation module generates a stroma mask to isolate the stromal tissue, which is then used to construct spatial patch graphs for the proposed deep-learning model.	88
6.2	Overview of the proposed model for identifying tumor-associated stroma in multi-modal prostate histopathology slides: The input patches are represented as spatial graphs and passed through a feature extractor. The patch embeddings are fed into a graph attention network (GAT) module to capture inter-patch relationships and refine the features with neighborhood consistency regularization (NCR) for handling noisy labels. The source discriminator serves as adversarial multi-modal learning (AML) module to predict data source (biopsy/whole-mount). The stroma classifier and source discriminator are trained simultaneously with the goal of successfully classifying tumor-associated stroma using multimodal-invariant features.	90

7.1	Overview of the proposed self-supervised learning model architecture. A VQGAN is utilized to learn a codebook of perceptually rich visual tokens from diverse pathology image datasets. This learned discrete codebook serves as the interface between the VQGAN and a latent Transformer. The Transformer models image composition through autoregressive prediction of the next image token.	98
7.2	Illustration of scale-aware positional encoding for WSIs. This approach improves upon the traditional Transformer’s positional encoding by incorporating resolution information, specifically by scaling the positional encodings according to the magnification level of the input image.	101
7.3	Illustration of the dilated attention mechanism for WSI modeling. This mechanism employs various attention patterns to capture both short-range and long-range dependencies. The number and hyperparameters of these patterns are adaptable based on the sequence length. This configuration enables the extension of a pretrained Transformer to model very long sequences, obtaining representations of entire gigapixel images without adding extra layers or increasing computational costs.	104
7.4	Images reconstructed by VQGAN and generated by latent Transformer. (1) First row: Original input images. (2) Second row: Images reconstructed from pretrained VQGAN using learned token codebook. (3) Third row: Source images where only the upper half is visible to the latent Transformer for image outpainting via autoregressive prediction of the next image token. (4) Fourth row: Images generated by pretrained latent Transformer by extending the third row source images.	117
7.5	Accuracy of linear probing experiments using features extracted from different Transformer layers for patch-level image classification tasks.	119

LIST OF TABLES

2.1	Overview of inductive biases in standard deep learning components, adapted from Battaglia et al. [BHB18].	11
3.1	The summary of model performance on repoDB dataset under ten-fold cross validation.	46
3.2	The results of novel predictions on repoDB dataset.	47
3.3	The summary of model performance on external dataset.	47
3.4	New candidate drugs ranked by prediction scores by BiFusion for breast carcinoma and Parkinson disease.	48
4.1	Depression prediction performance on clean datasets.	69
4.2	Depression prediction performance on noisy datasets.	69
5.1	Concordance index of models on predicting progression free survival for prostate cancer patients following radical prostatectomy.	84
6.1	Performance comparison with model variants. Results are averaged over 5 folds and shown in terms of mean value $\pm$ standard deviation.	94
7.1	Classification of tumor presence in the PCam dataset. Top: performance comparison of CNN models with fewer than 1 million trainable parameters. Middle: evaluation of state-of-the-art models with a larger parameter budget (several million). Bottom: performance of our proposed methods utilizing the minimum number of trainable parameters.	108

7.2	Classification of tumor malignancy in the BreakHis dataset, comparing patient recognition rates across test image subsets at various magnifications. The methods fall into two categories: magnification-specific methods, trained and tested separately for each magnification level; and magnification-independent methods, using a single unified model trained and tested on images from all magnifications.	109
7.3	Model performance on the GlaS challenge test set using the full training set. . .	112
7.4	Model performance on the GlaS challenge test set across varying training subsets. Results are based on inductive learning unless otherwise noted in parentheses. .	113
7.5	Model performance in segmenting prostate dataset images into "stroma" (BG), "benign" (BN), "low-grade" (LG), and "high-grade" (HG) categories.	114
7.6	mIOU on the prostate dataset across various training subsets. Results are based on inductive learning unless otherwise noted in parentheses.	114
7.7	Whole slide image classification results on Camelyon16	115

ACKNOWLEDGMENTS

I am deeply grateful to my Ph.D. advisor, Dr. Corey Arnold, for his unwavering support and insightful guidance throughout my doctoral journey. My appreciation also extends to my committee members—Dr. William Speier, Dr. Dan Ruan, Dr. Yingnian Wu, and Dr. Aydogan Ozcan—for their invaluable input and enthusiastic support.

I would like to acknowledge both the current and former members of the UCLA Biomedical Artificial Intelligence Research Lab and the Medical Imaging Informatics Group for their collaboration and support. Special thanks to my colleagues Wenyan Li, Jiayun Li, Haoyue Zhang, Jennifer Polson, Karthik Sarma, Johnny Ho, Yiwen Meng, Mara Pleasure, Katya Redekop, Carlos Olivares, Saarang Panchavati, and Alex Raman, among others. I am also profoundly thankful to our committed staff members Kimberly Flores, Rushi Kulkarni, Shawn Chen, Melanie Sobri and Rose Doherty for their dedicated support.

Lastly, my heartfelt thanks to the UCLA Bioengineering Department and the NIH for their financial support during my Ph.D. journey. I would also like to express my deepest gratitude to my friends and family, whose constant support and love have been my greatest strength.

VITA

2014–2018 B.E. (Information Engineering), Zhejiang University

2018-2020 M.S. (Bioengineering), UCLA

2021 Machine Learning Research Intern at IQVIA

2022 Machine Learning Research Intern at Genentech

SELECTED PUBLICATIONS

Wang, Z., Zhou, M. and Arnold, C.W., 2020. Toward heterogeneous information fusion: bipartite graph convolutional networks for in silico drug repurposing. *Bioinformatics*, 36(Supplement_1), pp.i525-i533.

¹ **Wang, Z.***, Li, J.*, Pan, Z., Li, W., Sisk, A., Ye, H., Speier, W. and Arnold, C.W., 2021. Hierarchical graph pathomic network for progression free survival prediction. *International Conference on Medical Image Computing and Computer-Assisted Intervention–MICCAI 2021*. Springer International Publishing.

Wang, Z., Pleasure, M., Zhang, H., Flores, K., Sisk, A., Speier, W. and Arnold, C.W., 2023. Deep Learning for Tumor-Associated Stroma Identification in Prostate Histopathology Slides. *International Conference on Medical Image Computing and Computer-Assisted Intervention–MICCAI 2023*. Cham: Springer Nature Switzerland.

¹*denotes equal contribution

Wang, Z., Pleasure, M., Redekop, E., Speier, W. and Arnold, C.W., 2024. Self-Supervised Learning for Digital Pathology using Vector-Quantized Image Modeling. In submission

Wang, Z., Redekop, E. and Arnold, C.W., 2024. Temporal Graph Learning using Heterogeneous Electronic Health Records to Predict Depression. In submission.

Wang, Z.*, Bazgir, O.*, Hafner, M. and Lu, J., 2024. Integration of Graph Neural Network and Neural-ODEs for Tumor Dynamic Prediction. ICLR 2024 Workshop.

Ding, K., Zhou, M., **Wang, Z.**, Liu, Q., Arnold, C.W., Zhang, S. and Metaxas, D.N., 2024. Graph Convolutional Networks in Medical Imaging: Concept, Architecture, and Clinical Application. ACM Computing Surveys (CSUR), under review.

Li, W., Li, J., Polson, J., **Wang, Z.**, Speier, W. and Arnold, C.W., 2022. High resolution histopathology image generation and segmentation through adversarial training. Medical Image Analysis, 75, p.102251.

Li, W., **Wang, Z.**, Yue, Y., Li, J., Speier, W., Zhou, M. and Arnold, C.W., 2020. Semi-supervised learning using adversarial training with good and bad samples. Machine Vision and Applications, 31, pp.1-11.

Li, W., **Wang, Z.**, Li, J., Polson, J., Speier, W. and Arnold, C.W., 2019. Semi-supervised learning based on generative adversarial network: a comparison between good GAN and bad GAN approach. CVPR 2019 Workshop.

CHAPTER 1

Introduction

Graphs are pervasive in biomedicine, from molecular interactions to patient networks. This prevalence stems from the natural propensity of biological patterns to be efficiently captured using graph structures [Vel23], providing a unified framework to describe a multitude of biomedical entities and their associations [LHZ22]. For instance, protein–protein interaction graphs use nodes to represent proteins and edges to denote physical interactions; patient electronic health records (EHR) can be modeled as graphs of medical codes and concepts generated through co-occurrences in longitudinal patient histories; and medical images can be conceptualized as spatial networks of cells and tissue regions.

Recent advancements in graph representation learning, particularly through the adoption of graph deep learning that extends convolutional neural networks (CNNs) to graph data, have been transformative. The introduction of graph convolutional networks (GCNs) [KW16] marked a new era in graph learning, significantly enhancing traditional CNNs’ capacity to process non-Euclidean data and achieving state-of-the-art results in various machine learning applications. Graph deep learning leverages fundamental principles established over the past two decades in systems biology and medicine [BGL11]. These principles correlate graph structures with molecular phenotypes, biological functions, and disease states [LHZ22], providing a theoretical basis that underpins the success of graph-based representation learning. Specifically, the local hypothesis suggests that interacting entities within a graph are more likely to exhibit similar characteristics than non-interacting ones [BGL11]. This implies that in protein interaction networks, for example, mutations in closely interacting proteins are

often associated with similar diseases. Similarly, the shared-components hypothesis indicates that diseases caused by disturbances in the same molecular or cellular components tend to exhibit similar phenotypes and respond similarly to treatments. These hypotheses extend to the analysis of multi-scale biomedical data, such as high-level patient medical histories and health statuses in EHRs. In this context, patients with similar medical codes, which represent comparable treatments or conditions, are likely to experience similar health outcomes. For instance, clusters of patients who share specific treatment codes might be at a common risk for certain diseases, illustrating the broader principle that similar interactions within a graph can predict comparable disease manifestations.

However, several challenges have been identified in this dissertation for further research:

- Biomedical systems often incorporate graph entities and interactions from various types and sources, forming heterogeneous graphs that are rich in complex structural and semantic content. A pivotal research question involves developing expressive graph representations that can effectively integrate this diverse information, which is essential for leveraging graph deep learning in biomedical research.
- Medical image analysis requires a comprehensive analysis of morphological, spatial, and topological data across different scales. Histopathology slides, for instance, are inherently gigapixel and hierarchical, featuring complex tissue characteristics including multi-scale patterns of cell nuclei, glands, and lymphocytes. Designing efficient graph construction methods for transforming images to graphs is crucial, as these methods greatly impact model performance, especially across different magnifications and multimodal data.
- The scarcity of labeled training data in medical imaging poses significant challenges. For example, labeling histopathology images is both costly and time-consuming. Pre-training graph models to serve as foundational models can improve their generalization abilities and facilitate efficient adaptation to downstream tasks. Exploring how to ap-

ply self-supervised learning to robustly represent graph entities and generalize edge connections remains an open and valuable research area.

1.1 Motivation

Throughout this dissertation, we address the aforementioned challenges of applying graph deep learning in biomedical research. Our approach begins by enhancing graph learning for heterogeneous data analysis, focusing on applications in drug repurposing and electronic health record (EHR) modeling across various biomedical scales. We then pivot to medical image analysis, aiming to develop efficient graph representation learning methods for histopathology image analysis. Additionally, we strive to enhance the robustness of image representations through a variety of techniques, ranging from handcrafted feature engineering and adversarial training to graph-based self-supervised learning. Ultimately, we introduce a unified pretraining framework designed to advance generalizable image representation learning in computational pathology.

1.1.1 Enhancing Graph Learning for Heterogeneous Information Fusion in Drug Repurposing and EHR Modeling

The increasing availability of large-scale biological databases has significantly enhanced pharmaceutical research. Drug repurposing (DR), which seeks new therapeutic uses for existing drugs, is particularly beneficial due to the high costs and extensive timelines associated with de novo drug development. However, drug repurposing encounters challenges due to the molecular heterogeneity of diseases and varied drug-disease associations. Previously, drug and disease representations were modeled on homogeneous graphs consisting solely of drugs, diseases, or targets. More advanced methods now incorporate multimodal data to construct heterogeneous graphs, which facilitate the aggregation of neighborhood information and improve predictions of drug-disease interactions. Consequently, drug repurposing requires an

in-depth exploration of multimodal biological networks within an integrative framework. A significant limitation of conventional GCN-based methods is their assumption of uniform node attributes across the graph, overlooking the unique features of nodes from different domains. This dissertation introduces a bipartite graph convolution network model designed to improve heterogeneous information fusion for drug repurposing.

Advancements in deep learning algorithms have also significantly improved disease modeling using EHR data, which offers longitudinal insights into patients’ medical histories and uncovers latent patterns. Nonetheless, the complex and irregular nature of clinical events recorded in EHRs presents substantial challenges, including data sparsity and heterogeneity. Existing deep learning models often overlook crucial temporal contextual information and fail to recognize potential collaborative signals among different patient sequences. Additionally, these models typically ignore domain-specific differences in codes from various sources. To overcome these challenges, this dissertation presents a temporal graph learning approach that dynamically models the evolving interactions and dependencies among EHR sequences. This model employs distinct learnable parameter sets for different data types, enabling the preservation of unique representation spaces while enhancing heterogeneous information fusion.

1.1.2 Improving Graph Learning for Multi-scale Pathology Image Analysis

The advent of digitized histopathological images provides an invaluable asset for supporting rapid and precise clinical decision-making. Whole slide images (WSIs) encompass a wealth of tissue characteristics observable across multiple scales. These images uniquely showcase extensive pathological traits and cellular interactions that are not discernible in other types of clinical imagery. Utilizing these patterns, we can construct biological graph networks that offer insights into disease states and help identify predictive imaging biomarkers.

Recent advancements in graph deep learning have proven particularly effective for addressing the complexities of histopathological applications, including data annotation, tissue

connectivity, and the integration of both global and local information. The graph structure effectively encapsulates tissue content connectivity across an entire slide, significantly reducing the need for labor-intensive labeling. This method captures contexts across multiple scales, improving disease outcome predictions by considering both comprehensive and detailed features. Moreover, it utilizes interactions among spatially-separated tiles, which enhances the model’s flexibility and expands its receptive field.

Despite these benefits, high-resolution histopathology images do not naturally conform to a graph structure, making efficient graph construction essential for model development and optimization. Current methodologies are typically divided into patch-based and cell-based approaches. Achieving a balance between efficiency and granularity in graph modeling remains a challenge, particularly due to the varying demands of different magnifications and modalities. In this dissertation, we investigate these challenges with a focus on improving graph deep learning techniques for prostate cancer histopathology, aiming to enhance patient-level survival prediction and tissue-level tumor microenvironment modeling.

1.1.3 Advancing Generalizable Image Representation Learning for Computational Pathology

Developing an optimal graph representation for computational pathology relies on two pivotal elements: a robust feature extractor for initializing node embeddings that encode discriminative image features, and a versatile image composition modeling module to build connections between nodes. Such graph structures are typically tailored to specific tasks, as different downstream applications demand unique configurations. A major hurdle in this field is the scarcity of data and the high costs associated with gathering annotations, which complicate the optimization of graph representations through fully-supervised learning. Self-supervised learning (SSL) has gained recognition as an effective approach for establishing fundamental data representations, thereby enhancing generalizability for downstream applications.

Motivated by recent breakthroughs in foundational models for natural image and lan-

guage processing, our dissertation presents a self-supervised learning framework tailored for computational pathology. We propose using vector-quantized modeling to encode a wide array of pathology phenotypes as a general graph node feature initializer through extensive pretraining. Using this domain-specific image tokenizer, we further develop a Generative Pretrained Transformer (GPT) to enhance generalizable image composition modeling. As detailed in the Background section 2.2.3, Transformer is a specialized form of attention-based Graph Neural Network (GNN). Conceptually, our approach employs a complete graph, allowing the Transformer to dynamically adjust the edge weights throughout the training and inference phases.

Further, pathology slide images differ from natural images primarily in their gigapixel and hierarchical structures. Existing SSL models for digital pathology often neglect these distinctions, overlooking magnification-specific information and lacking efficient strategies for adaptation across different magnification levels. In contrast, our proposed solutions enable the model to explicitly account for the magnification of input data in downstream tasks. We also investigate efficient finetuning strategies to adapt the pretrained model for various tasks across different magnification levels. We aim to establish our pretrained model as a versatile backbone, facilitating rapid adaptation for future research endeavors.

1.2 Contributions

The main contributions of this dissertation can be summarized in four specific aims: the first two aims concentrate on heterogeneous data analysis, while the last two focus on pathology image analysis.

Aim 1. Develop a bipartite graph convolutional network for drug repurposing through heterogeneous information fusion.

- a. To integrate multi-scale pharmaceutical information by constructing a multi-relational graph encompassing drug-protein, disease-protein, and protein-protein

interactions.

- b. To model inter-domain information fusion using bipartite graph convolution operations within an end-to-end graph learning framework.
- c. To enhance in silico drug repurposing by improving drug-disease association mining from heterogeneous biomedical knowledge graphs.

Aim 2. Develop a temporal graph learning approach that utilizes heterogeneous electronic health record (EHR) to improve disease modeling.

- a. To transform sequential EHR data into temporal graphs that capture dynamic patterns and high-order collaborative signals across various patient health trajectories
- b. To integrate heterogeneous graph convolution layers with a temporal attention mechanism, enabling effective aggregation of temporal-topological features from large-scale EHR graph datasets.
- c. To enhance patient depression prediction using EHR, facilitating early depression diagnosis for timely and personalized patient care.

Aim 3. Develop graph deep learning approaches for histopathology image analysis across multiple magnification levels.

- a. To develop a hierarchical graph convolutional network framework for prostate cancer survival prediction using whole slide images (WSIs), integrating both nuclei-level and patch-level graph structures.
- b. To characterize the prostate tumor-associated stroma through spatial graph modeling, enhancing the accuracy and efficiency of analyzing the tumor microenvironment in prostate cancer pathology.
- c. To explore the WSI representation learning using handcrafted features, self-supervised learning, and adversarial training.

- Aim 4.** Develop a self-supervised learning framework for computational pathology that unifies image generation and representation learning.
- a. To construct a domain-specific image tokenizer to encode diverse pathology phenotypes using vector-quantized image modeling.
 - b. To learn foundational WSI representations that support robust generalization across multiple magnifications using a scale-aware Generative Pretrained Transformer (GPT).
 - c. To develop efficient fine-tuning strategies that adapt the pretrained model for downstream tasks, facilitating the development of data-efficient deep learning models capable of transferring to diverse clinical workflows.

1.3 Outline of the Dissertation

This dissertation is organized as follows: Chapters 3 and 4 address Aims 1 and 2, respectively. Chapters 5 and 6 cover Aim 3, and Chapter 7 is dedicated to Aim 4.

- Chapter 2** provides the background on graph neural networks, Transformers, and self-supervised learning.
- Chapter 3** presents a bipartite graph convolutional network approach for heterogeneous information fusion in computational drug repurposing.
- Chapter 4** develops a temporal graph learning method that utilizes heterogeneous EHRs to enhance depression prediction.
- Chapter 5** designs a deep learning system employing hierarchical graph-based representations of WSIs for prostate cancer survival prediction.
- Chapter 6** presents a graph deep learning approach to characterize the tumor-associated stroma in multi-modal prostate histopathology slides.

Chapter 7 proposes a self-supervised learning framework for digital pathology that utilizes vector-quantized image modeling and a scale-aware GPT.

Chapter 8 concludes the dissertation and discusses limitations and potential future directions.

CHAPTER 2

Background

2.1 Graph Neural Network

2.1.1 Inductive Bias

Learning is the process of acquiring useful knowledge through observation and interaction with the world. A learning algorithm searches a solution space to find an option that better explains data or maximizes rewards [BHB18]. Inductive biases prioritize certain solutions over others, regardless of observed data, by enforcing constraints on the relationships and interactions within the learning process. These biases may reflect assumptions about the data generation process or the solution space [BHB18]. For instance, an $L2$ regularization term is an inductive bias that favors solutions with smaller parameter values; while the linear least squares method assumes that the data-generating process can be simply modeled as a linear function perturbed by additive Gaussian noise. Ideally, inductive biases enhance solution search efficiency and generalizability without significantly compromising performance.

Recent years have seen a surge in innovative deep learning architectures, typically built by assembling basic building blocks into complex, deep computational hierarchies. According to Battaglia et al. [BHB18], these architectures can be analyzed from a unified perspective by examining their inductive biases. As shown in Table 2.1, the building block of multilayer perceptrons (MLP) is the fully connected layer, which introduces a weak inductive bias allowing all input units to interact and determine the value of any output unit. Convolutional layers, stacked to form convolutional neural networks (CNN), differ from fully connected

Table 2.1: Overview of inductive biases in standard deep learning components, adapted from Battaglia et al. [BHB18].

Component	Entities	Relations	Inductive Bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

layers by imposing inductive biases of locality and translation invariance. These biases are particularly effective for processing natural image data, where there is high covariance within local neighborhoods that diminishes with distance. The recurrent layer, the building block of recurrent neural networks (RNN), reflects the relational inductive bias of temporal invariance (analogous to CNN’s translational invariance in space) and a bias for locality in the sequence via their Markovian structure. While these standard deep learning blocks contain various relational inductive biases and improve the learning process across diverse domains, there is no default deep learning component that operates on arbitrary relational structures, especially irregular graph-structured data. Specifically, CNNs are better suited for operating on regular Euclidean data like images (2D grids) and texts (1D sequences), which can be seen as special instances of graphs. Therefore, extending deep neural models to non-Euclidean domains has been an emerging research area. Graph deep learning, which supports arbitrary relational structures, has received enormous attention. In the next section, we will delve into the graph neural network (GNN), a framework for graph computations that offers a stronger inductive bias than that of convolutional and recurrent layers.

2.1.2 Computational Framework

2.1.2.1 Problem Formulation

Let $G = (V, E)$ denotes a graph, where $V = \{\mathbf{v}_i\}_{i=1}^{N^v}$ represents the set of nodes with cardinality N^v , and $E = \{(e_k, r_k, s_k)\}_{k=1}^{N^e}$ denotes the set of edges. Each edge is characterized by an attribute e_k , with r_k and s_k indicating the indices of the receiver (destination) and sender (source) nodes, respectively. The adjacency matrix $A \in \mathbb{R}^{N^v \times N^v}$ defines the connectivity between nodes:

$$a_{rs} = \begin{cases} 1 & (r, s) \in E \\ 0 & (r, s) \notin E \end{cases}$$

Node attributes are stored in the matrix $X^v \in \mathbb{R}^{N^v \times C^v}$, and edge attributes in $X^e \in \mathbb{R}^{N^e \times C^e}$, where C^v and C^e are the numbers of features for nodes and edges, respectively. The objective of graph neural networks is to learn effective node and edge embeddings, represented by $H^v \in \mathbb{R}^{N^v \times F^v}$ and $H^e \in \mathbb{R}^{N^e \times F^e}$ respectively, where F^v and F^e denote the dimensions of the node and edge embeddings. This learning process leverages both the graph structural information and the attributes of nodes and edges.

2.1.2.2 General Computational Steps

The fundamental concept of a GNN is to iteratively update node (or edge) representations by combining the embeddings of neighboring nodes and associated edges. A typical GNN consists of a series of propagation layers, where each layer l performs three computational steps. For simplicity, we will omit edge information in the following discussion, as it can be seamlessly integrated with node embeddings during information propagation at each step:

- **Message Passing:** the GNN computes neural messages $\mathbf{m}_{r,s}^{(l)} = \text{MSG}(\mathbf{h}_r^{(l-1)}, \mathbf{h}_s^{(l-1)})$ for linked nodes r and s based on their embeddings from the previous layers $\mathbf{h}_r^{(l-1)}$ and $\mathbf{h}_s^{(l-1)}$. This message passing mechanism varies depending on the type of graph.

For example, it is unidirectional for directed graphs and bidirectional for undirected graphs. The passing function MSG may use a uniform set of learnable parameters for homogeneous graphs, or it can have edge-specific configurations for heterogeneous graphs.

- **Message Aggregation:** The messages between node r and its neighboring nodes $\mathcal{N}_r$ are aggregated using $\hat{\mathbf{m}}_r^{(l)} = \text{AGG}(\mathbf{m}_{r,s}^{(l)} \mid s \in \mathcal{N}_r)$. This operation reflects the locality constraint similar to how a CNN operates over a small neighborhood of each image pixel. The aggregation function AGG is typically permutation invariant, accommodating the unordered nature of nodes in a graph.
- **Embedding Update:** The embeddings of the receiver nodes are updated by applying a nonlinear transformation: $\mathbf{h}_r^{(l)} = \text{UPD}(\hat{\mathbf{m}}_r^{(l)}, \mathbf{h}_r^{(l-1)})$. This update combines the aggregated message with the previous layer’s embedding. GNNs can capture complex and nonlinear patterns through multi-hop propagation across multiple layers of neural message passing.

2.1.2.3 Representative Models

Graph Convolutional Network (GCN) [KW16] is a widely adopted graph neural network architecture, renowned for its simplicity and effectiveness across various tasks and applications. It builds upon earlier work on spectral graph convolutions [DBV16], employing the following propagation rules:

$$H^l = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{l-1} W^{l-1} \right). \quad (2.1)$$

$\tilde{A} = A + \mathbf{I}$ is the adjacency matrix with self-connections ($\mathbf{I}$ is the identity matrix). $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. σ represents a nonlinear activation function such as ReLU. We can further dissect the above equation to examine the three computational functions defined in the GCN. For a node r , the above equation can be reformulated as

below:

$$\begin{aligned}
H_r^l &= \sigma \left(\sum_{s \in \{\mathcal{N}_r \cup r\}} \frac{\tilde{A}_{rs}}{\sqrt{\tilde{D}_{rr} \tilde{D}_{ss}}} H_s^{l-1} W^l \right) \\
H_r^l &= \sigma \left(\sum_{s \in \mathcal{N}_r} \frac{A_{rs}}{\sqrt{\tilde{D}_{rr} \tilde{D}_{ss}}} H_s^{l-1} W^l + \frac{1}{\tilde{D}_r} H_r^{l-1} W^l \right)
\end{aligned} \tag{2.2}$$

We can see that the message passing function MSG sends embeddings of neighboring nodes from previous layers. The message aggregation function AGG is defined as the weighted average of the neighbor node embeddings. The weight of the neighbor s is determined by the edge weight between r and s (i.e. A_{ij} normalized by the degrees of the two nodes). The message update function UPD combines the aggregated messages with the node's own embedding, normalized by its degree, and applies a nonlinear activation function.

GraphSAGE [HYL17] is a general inductive framework that generates node embeddings by sampling and aggregating features from a node's local neighborhood. It utilizes node feature information to efficiently produce embeddings for previously unseen data, employing the following propagation rules:

$$\mathbf{h}_r^l = \sigma(\mathbf{W}^l \cdot \{\text{AGG}(\mathbf{h}_{r,s}^{(l-1)} \mid s \in \mathcal{N}_r)\} \parallel \{\mathbf{h}_r^{l-1}\}). \tag{2.3}$$

$\mathbf{W}^l$ represents the weight matrices for linear projection, and $\parallel$ denotes the concatenation operation. GraphSAGE utilizes three types of aggregation functions: mean aggregator, LSTM aggregator, and max pooling aggregator. Specifically, GraphSAGE with a mean aggregation function can be viewed as an inductive version of GCN. The concatenation operation acts as a form of "skip connection" between the different layers of the GraphSAGE model, helping to mitigate the over-smoothing issue and enhance model performance as layer depth increases.

Graph Attention Network (GAT) [VCC17] employs an attention mechanism to dynamically assign weights to neighboring nodes, enhancing the stability of the learning process and mitigating noise effects. One advantage of attention mechanisms is their ability to focus on the most relevant parts of the input for decision-making. GAT introduces a computationally

efficient attentional layer that utilizes self-attention and multi-head attention mechanisms. This layer is parallelizable across all nodes in the graph, allows for differential weighting of nodes based on their connectivity, and operates independently of the overall graph structure. The attention coefficients and the propagation process in GAT are described as follows:

$$\alpha_{rs} = \frac{\exp(\sigma(\alpha^T [W\mathbf{h}_r^{l-1} \| W\mathbf{h}_s^{l-1}]))}{\sum_{k \in \mathcal{N}_r} \exp(\sigma(\alpha^T [W\mathbf{h}_r^{l-1} \| W\mathbf{h}_k^{l-1}]))} \quad (2.4)$$

$$\mathbf{h}_r^l = \sigma(\sum_{s \in \mathcal{N}_r} \alpha_{rs} W \mathbf{h}_s^{l-1}) \quad (2.5)$$

where α is the weight vector of a single-layer MLP, W is the weight matrix for the linear transformation applied to each node, and T denotes transposition.

2.1.3 Design Pipeline

2.1.3.1 Design Graph Structure

First, it is essential to identify the graph structure relevant to the learning task. This involves considering two main scenarios: structural and non-structural. In structural scenarios, the graph structure is explicitly defined within the task, such as in applications dealing with molecules, physical systems, knowledge graphs, etc. Conversely, in non-structural scenarios, the graph structure is implicit and must be constructed from the task at hand. For example, in medical imaging, such as histopathology, it is necessary to define graph nodes (e.g., cells or tissue regions based on the required resolution for the task) and the edges between them. Edges might be determined by spatial distances, under the assumption that entities closer to each other are more likely to interact.

Upon establishing the fundamental elements of the graph structure, further specifications regarding the graph types are needed. Graphs can typically be categorized as follows:

- **Directed/Undirected Graphs:** Directed graphs have all edges oriented from one

node to another, providing more structural information than undirected graphs.

- **Homogeneous/Heterogeneous Graphs:** In homogeneous graphs, all nodes and edges are of the same type, whereas heterogeneous graphs feature different types of nodes and edges.
- **Static/Dynamic Graphs:** Graphs are considered dynamic when either the input node/edge features or the topology changes over time. Static graphs maintain the same topological structure and initial node/edge features throughout the model training process.

2.1.3.2 Define Learning Task

Next, we need to establish the training settings based on the learning task type. Graph learning typically encompasses three types of tasks:

- **Node-level** tasks concentrate on individual nodes, including node classification, node regression, and node clustering. Node classification involves categorizing nodes into distinct classes using a shared classifier applied directly to $\mathbf{h}_r$ to predict targets for each node $u \in V$. Node regression involves predicting a continuous value for each node, while node clustering aims to partition nodes into several disjoint groups, grouping similar nodes together.
- **Edge-level** tasks, such as edge classification and link prediction, require the model to classify types of edges or to predict the existence of an edge between two specified nodes. For this, a classifier is trained using the combined features of two nodes, $\mathbf{h}_r \parallel \mathbf{h}_s$, possibly including edge-level attributes.
- **Graph-level** tasks include graph classification, graph regression, and graph matching, which require the model to learn comprehensive graph representations. To predict targets for an entire graph, an invariant output is necessary. This is achieved by

aggregating all node embeddings $\mathbf{h}_r$ into a single graph-level representation, e.g., by using $\text{AGG}(\mathbf{h}_r \mid r \in V)$, followed by training a classifier on this unified vector.

2.1.3.3 Build Computational Modules

Finally, we can construct models using the graph computational blocks described below:

- **Propagation Module:** This module propagates neural messages between nodes to integrate feature and topological information. As outlined in Section 2.1.2.3, typical implementations involve standard graph convolutional modules or attentional graph modules for aggregating neighbor information, supplemented by skip connections to address the over-smoothing problem in deep graph models.
- **Sampling Module:** Depending on the computational resources available (e.g., GPU speed and memory), this module is crucial for large-scale graphs. It enables efficient propagation by reducing the neighborhood size, either by selecting a subset of each node’s neighbors as done in GraphSAGE [HYL17], or by sampling multiple subgraphs and limiting neighborhood searches within these, as in ClusterGCN [CLS19]. This module is typically integrated with the propagation module to enhance efficiency.
- **Pooling Module:** This module is essential for deriving representations of high-level subgraphs or entire graphs by consolidating information from multiple nodes. Similar to the process in computer vision where a convolutional layer precedes a pooling layer to abstract more generalized features, graphs—particularly complex, large-scale ones—contain rich hierarchical structures crucial for node-level and graph-level classification tasks. Previous research in this area includes direct pooling methods and more sophisticated hierarchical pooling strategies. For instance, SortPooling [ZCN18] sorts node embeddings based on the structural roles of nodes, feeding these sorted embeddings into CNNs to obtain the final representation. DiffPool [YYM18] introduces a learnable hierarchical clustering module that generates an assignment matrix

at each layer to guide pooling. Additionally, SAGPool [LLK19] employs a self-attention mechanism to jointly utilize features and topology for learning graph representations, optimizing both time and space complexity.

2.2 Transformer

2.2.1 Original Model Architecture

The original Transformer, introduced by Vaswani et al. [VSP17], is a sequence-to-sequence model comprising an encoder and a decoder, each built from L identical layers. Each encoder layer primarily consists of a multi-head self-attention module and a position-wise feed-forward network (FFN). To facilitate deeper models, each module includes a residual connection [HZR16] followed by layer normalization [BKH16a]. Unlike encoder layers, decoder layers incorporate an additional cross-attention module positioned between the multi-head self-attention modules and the position-wise FFNs. Additionally, the decoder’s self-attention modules are modified to prevent forward-looking attention, ensuring that each position can only attend to past positions. Figure 2.1 depicts the architecture of the original Transformer. The key modules of this architecture are further detailed in the following subsection.

2.2.1.1 Attention Modules

The Transformer architecture employs an attention mechanism based on the Query-Key-Value (QKV) model. The mechanism involves packed matrix representations for queries $\mathbf{Q} \in \mathbb{R}^{N \times D_k}$, keys $\mathbf{K} \in \mathbb{R}^{M \times D_k}$, and values $\mathbf{V} \in \mathbb{R}^{M \times D_v}$. The scaled dot-product attention is formulated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right) \mathbf{V} = \mathbf{A}\mathbf{V} \quad (2.6)$$

Here, N and M represent the lengths of the queries and keys/values respectively, while D_k and D_v are their respective dimensions. The matrix $\mathbf{A}$, termed the attention matrix, is

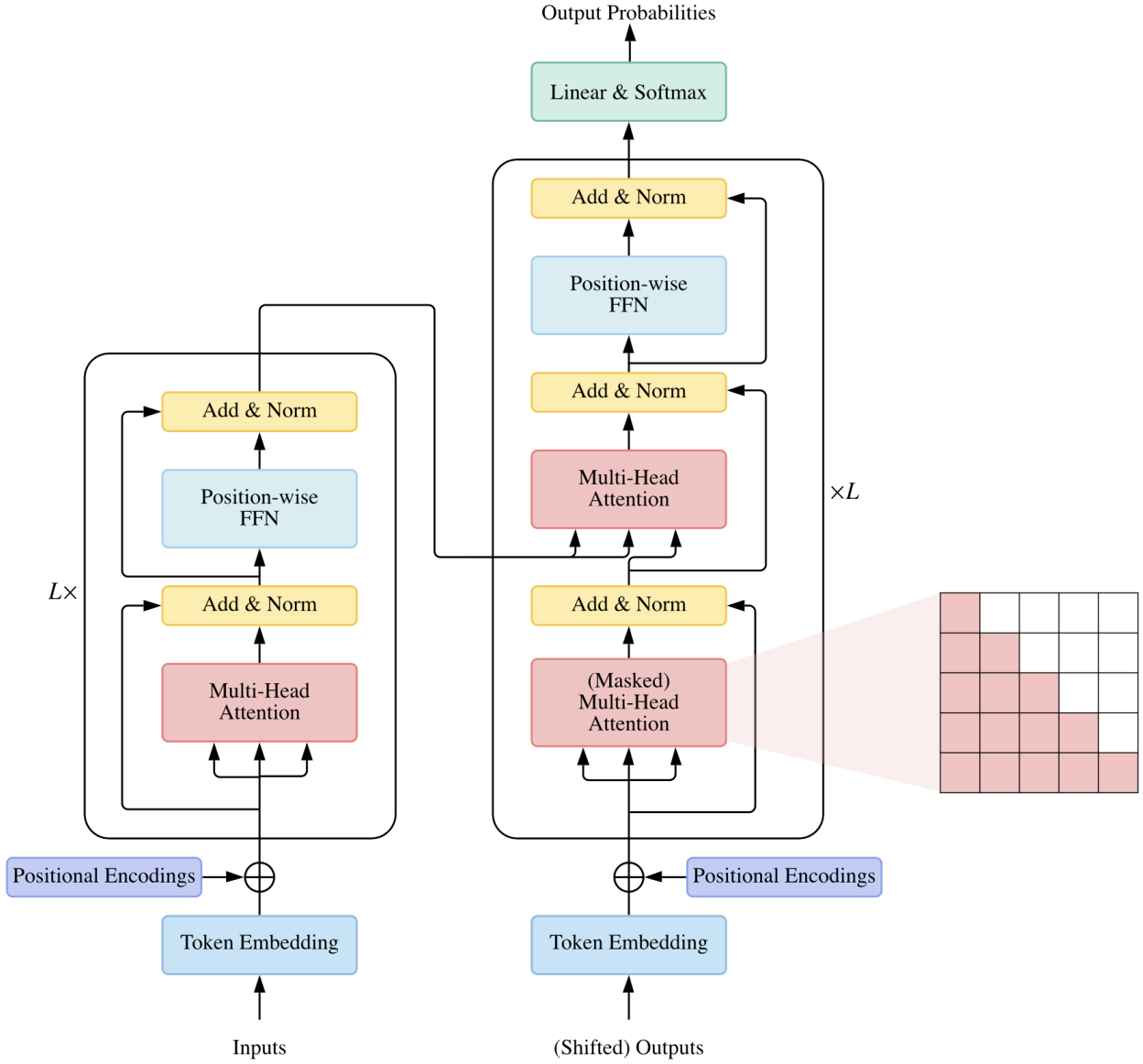


Figure 2.1: Overview of the original Transformer architecture, reproduced from [LWL22].

computed as $\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)$ applied row-wise. This normalization mitigates the vanishing gradient issue of softmax function.

To enhance stability and performance, the Transformer leverages multi-head attention. This approach involves projecting the original D_m -dimensional queries, keys, and values into

H distinct sets of D_k and D_v dimensions:

$$\text{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) \mathbf{W}^O, \quad (2.7)$$

$$\text{where head}_i = \text{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V). \quad (2.8)$$

Transformer differentiates three types of attention based on the source of queries and key-value pairs:

- **Self-attention:** Employed in the encoder, this type sets $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$ from the output of the previous layer.
- **Masked Self-attention:** Used in the decoder, it restricts attention such that positions only attend to preceding and current key-value pairs. A mask is applied to the pre-normalized attention scores $\hat{\mathbf{A}} = \exp\left(\left(\frac{\mathbf{QK}^T}{\sqrt{D_k}}\right)\right)$ to enforce causality by setting $\hat{A}_{ij} = -\infty$ for $i < j$.
- **Cross-attention:** The decoder's queries come from its prior layer outputs, while keys and values are derived from the encoder outputs.

2.2.1.2 Position-wise FFN

The position-wise Feed-Forward Network (FFN) is a fully connected module that processes each position independently with the same operations. It is mathematically defined as:

$$\text{FFN}(\mathbf{H}') = \text{ReLU}(\mathbf{H}'\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2, \quad (2.9)$$

where $\mathbf{H}'$ represents the output from the preceding layer. The trainable parameters include $\mathbf{W}^1 \in \mathbb{R}^{D_m \times D_f}$ and $\mathbf{W}^2 \in \mathbb{R}^{D_f \times D_m}$ for weights, and $\mathbf{b}^1 \in \mathbb{R}^{D_f}$ and $\mathbf{b}^2 \in \mathbb{R}^{D_m}$ for biases. Typically, the intermediate dimension D_f is chosen to be larger than D_m .

2.2.1.3 Residual Connection and Normalization

To construct a deep model, the Transformer utilizes a residual connection [HZR16] around each module, complemented by Layer Normalization [BKH16a]. Specifically, each encoder block in the Transformer can be expressed as follows:

$$\mathbf{H}' = \text{LayerNorm}(\text{SelfAttention}(\mathbf{X}) + \mathbf{X}), \quad (2.10)$$

$$\mathbf{H} = \text{LayerNorm}(\text{FFN}(\mathbf{H}') + \mathbf{H}'). \quad (2.11)$$

$\text{SelfAttention}(\cdot)$ represents the self-attention module, and $\text{LayerNorm}(\cdot)$ indicates the layer normalization process. Layer normalization adjusts and scales the features post-attention to ensure consistent magnitudes across different features, facilitating stable training across deeper network layers.

2.2.1.4 Position Encodings

Since the Transformer lacks recurrence and convolution, it does not inherently capture positional information, which is crucial for understanding sequence order, particularly in the encoder. To address this, the original Transformer model incorporates sinusoidal position encodings. For each position index t , the encoding is a vector $\mathbf{p}_t = \text{PE}(t) \in \mathbb{R}^{D_m}$, where each component is derived from sinusoidal functions of the position index at predetermined frequencies.

$$\text{PE}(t)_i = \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases} \quad (2.12)$$

where ω_k is the frequency for each dimension:

$$\omega_k = \frac{1}{10000^{2k/d}} \quad (2.13)$$

Derived from the function's definition, the frequencies within the sinusoidal positional encoding decrease along the vector dimension, forming a geometric progression in terms of

wavelengths, ranging from 2π to $10000 \cdot 2\pi$. A key feature of this encoding is that for any fixed offset k , the PE_{pos+k} can be represented as a linear function of PE_{pos} . This property significantly enhances the model’s ability to effortlessly attend to relative positions.

2.2.2 Model Usage

The Transformer architecture can be employed in three primary configurations:

- **Encoder-Decoder.** This configuration uses the complete Transformer architecture as discussed in the previous Section 2.2.1. It is commonly applied in sequence-to-sequence tasks such as neural machine translation [VSP17].
- **Encoder only.** In this setup, only the encoder component is used. The outputs serve as representations of the input sequence, typically for tasks in natural language understanding like text classification and sequence labeling. A notable example is BERT (Bidirectional Encoder Representations from Transformers) [DCL18].
- **Decoder only.** This configuration employs only the decoder, excluding the encoder-decoder cross-attention module. It is primarily used for sequence generation tasks, including language modeling. Key implementations include the GPT series (Generative Pretraining Transformer) [RNS18, RWC19, BMR20].

2.2.3 Connections with Graph Neural Network

The Transformer can be conceptualized as a GNN defined over a fully connected graph, with each input token acting as a node. Unlike traditional GNNs that incorporate structural knowledge about the data, the Transformer’s information propagation relies solely on similarity measures derived from token embeddings. This positions the Transformer as a variant of attentional GCN, where the architecture’s minimal assumptions about data structure afford a high degree of flexibility and universality.

In scenarios where the structure of the input data is not predefined, it is practical to model the data as a fully connected graph. This general approach aligns perfectly with the Transformer’s implementation, which inherently considers the input sequence as such a graph. As discussed in the previous section (see Section 2.1.2.2), each block within the Transformer can be interpreted as a set of graph information propagation layers. Initially, the message-passing function broadcasts each token’s embedding across all other tokens. Subsequently, the self-attention mechanism aggregates these messages using a dot-product operation. The integration of residual connections, layer normalization, and the FFN collectively function as mechanisms for updating node embeddings.

However, this lack of inherent structural bias can make the Transformer prone to overfitting, particularly when dealing with small datasets. Consequently, the Transformer is predominantly utilized in large-scale data modeling tasks, especially during model pretraining in self-supervised learning settings. This approach helps to mitigate overfitting by leveraging extensive data, thereby enhancing the model’s generalization capabilities.

2.3 Self-supervised Learning

The principal challenge in applying deep learning to biomedicine is its reliance on large-scale annotated datasets. Biomedical data are particularly difficult to label, often leading researchers to first train models on broad datasets like ImageNet, and then fine-tune them for specific tasks. However, this transfer learning strategy can be problematic, as initial training on non-biomedical tasks might introduce irrelevant features, reducing the models’ effectiveness for biomedical purposes. Conversely, self-supervised learning provides a more focused initial training by leveraging biomedical data directly, even without explicit labels. This approach better aligns the learned features with the needs of biomedical applications.

Self-supervised learning harnesses the inherent co-occurrence relationships within data as an inductive bias for learning, which offers flexible applications. This approach can be

divided into two primary categories:

- **Contrastive:** The goal is to train the model to recognize and associate similar samples while distinguishing dissimilar ones. This involves training an encoder to convert the input x into an explicit vector z that reflects similarity, using methods such as mutual information maximization or instance discrimination.
- **Generative:** This strategy focuses on understanding and reproducing the input data by predicting some aspects of the data from other parts. Specifically, it involves training an encoder to encode input x into a vector z , and a decoder to reconstruct x from z , applicable in scenarios like masked image reconstruction and next-token prediction.

2.3.1 Contrastive Learning Approaches

Recent advances in contrastive learning, exemplified by Deep InfoMax [HFL18], MoCo [HFW20], and SimCLR [CKN20], highlight the effectiveness of discriminative models in representation learning. The core principle of contrastive learning is to "learn to compare," employing a Noise Contrastive Estimation (NCE) [GH10] objective, which is structured as follows:

$$\mathcal{L} = \mathbb{E}_{x, x^+, x^-} \left[-\log \left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}} \right) \right] \quad (2.14)$$

where x^+ is similar to x , x^- is dissimilar to x and f represents an encoder (representation function). Although the specific similarity measure and encoder can differ depending on the task, the underlying framework remains consistent.

We can further categorize recent contrastive learning frameworks into two types: context-instance contrast and instance-instance contrast. The context-instance, or global-local contrast, focuses on modeling the relationship between a sample's local features and its overall context representation. As an alternative, instance-instance contrastive learning examines

the relationships between local representations of different samples at the instance level.

2.3.2 Generative Learning Approaches

Generative self-supervised learning seeks to develop representations that accurately reflect the underlying structure and distributions of data. This typically involves learning to generate or reconstruct the data from its corrupted or incomplete versions. Prominent techniques include autoregressive models and autoencoder models.

Autoregressive (AR) models, conceptualized as "Bayes net structures" (directed graph models), represent the joint distribution of data as a product of conditional distributions:

$$\max_{\theta} p_{\theta}(\mathbf{x}) = \sum_{t=1}^T \log p_{\theta}(x_t \mid \mathbf{x}_{1:t-1}) \quad (2.15)$$

where probability of each variable depends on its predecessors. In natural language processing, autoregressive language models typically aim to maximize the likelihood using a forward autoregressive factorization. For instance, GPT models leverage Transformer decoder architectures [RNS18, RWC19, BMR20]. Autoregressive models are also utilized in computer vision, exemplified by Pixel-CNN [OKE16] and iGPT [CRC20], where the approach involves modeling images pixel by pixel, with lower (right) pixels conditioned on the upper (left) pixels.

Autoencoder (AE) models aim to reconstruct inputs from their corrupted versions. Due to its versatility, the auto-encoding model is among the most prevalent generative models, with several variants. The first notable variant is the denoising autoencoder, which is designed to ensure that the representations are resilient to noise introduction. This concept is exemplified in natural language processing by the masked language model (MLM), such as BERT [DCL18], where random tokens are masked during training to be predicted based on their context information. The second variant is the variational autoencoder, which posits that data originate from latent variables. It uses variational inference to approximate the posterior distribution of these latent variables $Z = \{z_1, z_2, \dots, z_n\}$ given the data X is ap-

proximated by a variational distribution $q(z|x) \approx p(z|x)$. Training involves maximizing the evidence lower bound (ELBO) on the data’s log-likelihood:

$$\log p(x) \geq -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{q(z|x)}[\log p(x|z)] \quad (2.16)$$

where $p(x)$ is evidence probability, $p(z)$ is prior and $p(x|z)$ is likelihood probability. Variational Autoencoders (VAE) [KW13] assumes Gaussian distributions for both the prior $p(z)$ and the approximate posterior $q(z|x)$, specifically $p(z) \sim \mathcal{N}(0, 1)$. The reparameterization trick facilitates modeling the posterior $q(z|x)$, with $z = \mu + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$, and both μ and σ are outputs from neural networks. The decoder then reconstructs the input from the latent variable z . Additionally, the VQ-VAE [VV17] introduces a method for learning discrete latent variables, useful in modalities like language and images. It employs vector quantization to approximate the distribution of these variables, using a shared and learnable embedding table for nearest neighbor calculations.

Adversarial representation learning combines aspects of both generative and contrastive approaches, as seen in the development of Generative Adversarial Networks (GAN) [RMC15]. In GANs, the training process is conceptualized as a game between two components: a generator (G) that creates synthetic samples and a discriminator (D) that attempts to differentiate these from real samples. The objective is framed as a min-max game:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (2.17)$$

Adversarial learning aims to approximate the original data distribution rather than individual samples by minimizing distributional divergence. Unlike contrastive methods, which use only an encoder, adversarial learning maintains both an encoder and a decoder, enhancing its expressive capacity typical of generative models but also complicating the learning objective and potentially leading to unstable training dynamics. The presence of a decoder in adversarial settings requires the representation to be comprehensive, encompassing all details necessary to reconstruct the input. In contrast, contrastive methods focus solely

on capturing distinct features that differentiate samples. Overall, adversarial methods integrate strengths of both generative and contrastive techniques, making them suitable for tasks involving implicit distribution modeling, despite their inherent challenges.

2.3.3 Masked Modeling

Denoising forms a central theme in the current trend of self-supervised learning, with masked modeling serving as a key subtype. Noise in this context is introduced either through masking—predicting missing text tokens in language models like BERT [DCL18], or missing patches in images, as seen with Masked Autoencoders (MAE) [HCX22]. Alternatively, noise can also be additive Gaussian, where no clean signal is explicitly specified, exemplified by Denoising Diffusion Models (DDM) [HJA20].

Masked image modeling (MIM), originating from the widely adopted masked language modeling (MLM) paradigm in natural language processing, demonstrates promising results in visual representation learning. The MIM framework comprises four core modules: a Mask Strategy Module for generating mask sets, a Target Module for determining supervisory signals, an Encoder for pretraining with adaptable network architectures, and a Head Module for computing losses between supervisory signals and predictions. MAE [HCX22] and iGPT [CRC20] represent two foundational frameworks in MIM research [LZW23]: iGPT, similar to GPT, employs an autoregressive paradigm, whereas MAE, akin to BERT, utilizes a masked autoencoder paradigm.

Recent research highlights that the choice of target module for extracting various supervision signals influences model preferences. For instance, MAE [HCX22] and SimMIM [XZC22] focus on raw pixel for image reconstruction, while BEiT [BDP21] relies on a pre-trained tokenizer. Additionally, iBOT [ZWW21] formulates MIM as a knowledge-distillation task, employing a teacher-student framework with an online tokenizer. Building upon BEiT’s framework, BEiT_{v2} [PDB22b] utilizes vector-quantized knowledge distillation to discretize a continuous semantic space to compact codes. MAGE [LCM23] introduces a masked gener-

ative encoder, unifying representation learning and image synthesis, by employing semantic tokens learned by a vector-quantized GAN (VQGAN) at inputs and outputs. This approach demonstrates superior performance compared to operating on raw pixels for both generation and representation learning tasks.

CHAPTER 3

Bipartite Graph Convolutional Networks for In Silico Drug Repurposing

3.1 Abstract

Mining drug-disease association and related interactions are essential for developing in silico drug repurposing methods and understanding underlying biological mechanisms. Recently, large-scale biological databases are increasingly available for pharmaceutical research, allowing for deep characterization for molecular informatics and drug discovery. However, drug repurposing is challenging due to the molecular heterogeneity of disease and diverse drug-disease associations. Importantly, the complexity of molecular target interactions such as protein-protein interaction (PPI) remains to be elucidated. Drug repurposing thus requires deep exploration of a multi-modal biological network in an integrative context.

In this study, we propose BiFusion, a bipartite graph convolution network model for drug repurposing through heterogeneous information fusion. Our approach combines insights of multi-scale pharmaceutical information by constructing a multi-relational graph of drug-protein, disease-protein and protein-protein interactions. Especially our model introduces protein nodes as a bridge for message passing among diverse biological domains, which provides insights into utilizing PPI for improved drug repurposing assessment. Unlike conventional graph convolution networks always assuming the same node attributes in a global graph, our approach models inter-domain information fusion with bipartite graph convolution operation. We offered an exploratory analysis for finding novel drug-disease as-

sociations. Extensive experiments showed that our approach achieved improved performance than multiple baselines for drug repurposing analysis.

3.2 Introduction

Drug repurposing (DR) is a strategy to identify novel therapeutic purposes for existing drugs with a goal to expand the scope of the original medical indication of known drugs [LZC15]. This task is of great pharmaceutical significance since the de novo drug discovery is known to be costly and lengthy. The total cost of developing a drug ranges from \$2 billion to \$3 billion and it takes at least 13–15 years to bring a single drug to market [YYW18]. By contrast, drug repurposing offers a fast and cost-effective means for drug candidate discovery. For example, the repurposed drug candidate has proven to be sufficiently safe through preclinical assessments, thus resulting in a shortened period of clinical evaluation.

Recently, large-scale databases such as protein-protein interaction (PPI) networks, drug-target interactions and drug-disease associations are rapidly growing and increasingly accessible. The wealth of drug-related data presents great opportunities to generate novel insights surrounding drug mechanisms and develop in silico drug repurposing methods to accelerate drug discovery. However, in silico drug repurposing is challenging due to the molecular heterogeneity of disease and diverse drug-disease associations. For example, the complexity of molecular target interactions such as PPI remains to be elucidated. Without knowledge from a broader network of the molecular determinants of disease and drug targets, we are unable to develop efficacious drug treatment for complex diseases [GL17]. Therefore, drug repurposing requires deep exploration of a multi-modal biological network including drug-disease, drug/disease-protein, and protein-protein associations in an integrative context.

Among various biomedical interactions, the importance of the complex PPI network is broadly recognized in biological systems and the development of disease states [SBA16]. In other words, PPIs are at the center of almost every cellular process from cell motility to DNA

replication. Thus understanding PPI mechanism could greatly help elucidate the function of a known or novel protein and its role in a known biological pathway, which can be a key factor for justifying drug repurposing. Existing approaches using PPIs are often focused on using predefined descriptors to represent protein information such as the overlap of PPI closeness. However, they can not fully explore the potential information within PPI in an integrative context, which requires modeling inter-domain information fusion to characterize drug’s pharmacological action and guide a roadmap for drug repurposing assessment.

Previous studies in drug repurposing were primarily focused on drug and disease activities to uncover statistical associations between them. [DIA12, SAB12, YLW14]. These analyzes worked on a single data modality such as gene expression or drug-target interactions, only capturing partial information of the heterogeneous network. Additionally, these methods cannot consider the important topological information among different biological networks. To address this limitation, there has been a growing number of efforts to incorporate various data sources for boosting the accuracy of drug repositioning [GSR11, LL12, NZM13, ZAO13, LWL16, GMV16, ZZL19]. These approaches integrate multiple information sources such as chemical fingerprint, interaction network closeness of drug targets and correlation between drug side effects. However, these approaches with engineering-based features were unable to capture graph-structured information such as chemical molecules of drugs and PPI network knowledge. Therefore, graph-based model architectures become highly desired for incorporating multi-scale, graph-based knowledge and improving the model performance in drug repurposing.

Graph convolutional networks (GCNs) [KW16] extend deep learning approaches specifically designed for processing graph-based data in various graph-related tasks. In principle, GCNs perform a convolution by aggregating the neighbor nodes’ information to learn node representations in the entire graph. GCN has unique advances in its automation on feature extraction from raw graph-structured inputs. Recently, a handful of GCN-based methods in drug-drug and drug-target interaction predictions have proved the usefulness of GCN-based

models for drug information extraction [ZAL18, MXZ18, GFL18, FDC18]. However, exploring GCN-based networks for deep understanding multi-scale biological characteristics of drug data remains to be fully elucidated. Specifically, a notable limitation for conventional GCN-based methods is that same node attributes should always be assumed. Thus these approaches view the multi-relational networks as a global graph, completely ignoring distinct node features from different domains. For example, node features in drug and protein domains actually follow separate statistical distributions. But conventional GCNs can only leverage neighboring nodes in a single graph thus are quite difficult to measure correlation between the two separate domains.

To address all challenges above, we propose a bipartite graph convolution network approach, termed BiFusion, for *in silico* drug repurposing. The key motivation is to model interactions between diverse biological domains through bipartite graphs. Unlike previous GCN-based methods, our model enables inter-domain information fusion with a bipartite graph convolution operation. To allow information fusion, our model learns to represent different features from heterogeneous nodes into a unified embedding space, where protein nodes serve as a bridge for message passing within complex biological networks. This design differs from conventional GCN that is limited on node representation in a single graph. Overall, our major contributions can be summarized as follows: (i) To the best of our knowledge, this paper proposed the first bipartite GCN-based approach for *in silico* drug repurposing, assembling interactions across protein, drug and disease domains from large-scale databases. (ii) We proposed a novel end-to-end graph learning framework that can effectively integrate multi-relational interaction data for drug repurposing, yielding improved performance than baseline methods. (iii) Our analysis provided insights into better extracting and fusing information from the PPI network for drug repurposing.

3.3 Related Work

We briefly review computational approaches for drug repurposing and related research on graph convolutional networks.

Numerous studies have used single data source to identify drug indications. Different information modalities include structural features of compounds (or proteins) [DIA12], genetic activities [SAB12] and phenotypic profiles such as side effects of drugs [YLW14]. However, these methods failed to offer an unbiased perspective for predicting drug-disease associations due to the potential noise in the single information source. Additionally, these methods can not model important topological information among different biological networks. In response, current methods can be categorized into similarity-based and network-based approaches.

Most of similarity-based approaches are integrative methods using the heterogeneous information [GSR11, LL12, NZM13, ZAO13]. They rely on the assumption that similar drugs are indicated for similar diseases. These methods utilize shared characteristics between drugs such as drug-targets, chemical structures and adverse effects, and then constructed similarity features to build computational models. For example, PREDICT [GSR11] is a similarity-based framework integrating drug-drug similarity (based on drug-protein interactions, sequence and gene-ontology) and disease-disease similarity (disease-phenotype and human phenotype ontology), authors used them as key features applying logistic regression to predict similar drugs for similar diseases.

Network-based approaches [LWL16, GMV16, CDH18, CKB19, ZZL19] model graph-structured information among different biological networks to boost the performance for drug repurposing. Typically, in these models the nodes in the networks represent either drug, disease, or gene products and edges denote the interactions or relationships among them. For example, [CDH18] identified hundreds of new drug-disease associations by quantifying the network proximity of disease genes and drug targets in the human protein-protein

interactome. The deepDR [ZZL19] learnt high-level features of drugs from the heterogeneous networks by a multi-modal deep autoencoder and applied a variational autoencoder to infer candidates for approved drugs. However, deepDR considered information sources in the drug domain only without interactions in the disease domain.

Bipartite graph comprises a set of nodes decomposed into two disjoint sets [PKP18, AGC07], which is a natural representation for modeling complex items of biological systems and their interactions. Extensive studies have revealed the feasibility of bipartite graphs and their impact in the field of network biology [PKP18]. For example, [AGC07] built a bipartite graph to analyze relationships between drug targets and disease-gene products. [KPD16] performed a bipartite graph approach to analyze the relationships between human genetic diseases. In the field of drug repurposing, [LL12] developed a bipartite drug–target network method using drug pair similarity integrated drug chemical structure similarity, common drug targets and their interactions. [ZMW19] also constructed a bipartite graph model with known relationships between drugs and their target proteins. However, most of these methods heavily relied on predefined drug similarity features and ignored the important information sources in the disease domain. Although they utilized the PPI information, the relationships between drug targets and disease-gene products in the context of biological interaction network have not been investigated.

Graph convolutional network (GCN) [KW16] has opened a new paradigm for graph learning and achieved leading performance in machine learning tasks. Traditional graph-based approaches are inflexible at scale since they often rely on hand-engineered features including summary graph statistics and kernel functions. By contrast, GCN-based models are designed to capture the dependence of graphs via a recursive neighborhood aggregation scheme, where each node aggregates feature vectors of its neighbors to update its new node features. Thus GCNs demonstrate their superiority for graph-related tasks given their ability to naturally integrate the feature attributes of graph-structured data and learn intrinsic features from raw graph-structured inputs.

GCN extends the idea of the graph neural network (GNN) [SGT08]. Specifically, a graph $\mathcal{G} \in \mathbb{G}$ can be denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consisting of a vertex set $\mathcal{V} = \{v_i\}_{i=1}^{N_{\mathcal{V}}}$ and edge set $\mathcal{E} = \{e_j\}_{j=1}^{N_{\mathcal{E}}}$, $\vec{h}_{v_i} \in \mathbb{R}^N$ is the node feature of vertex v_i . A general GNN layer can be defined as follows:

$$g_{\mathcal{E}}(v_i) = \rho \left(\vec{h}_{v_i}, \text{agg} \left(\left\{ W_{v_i, v_j} \vec{h}_{v_j} \mid v_j \in \mathcal{N}_{v_i}^{\mathcal{E}} \right\} \right) \right), \quad (3.1)$$

where $W \in \mathbb{R}^{M \times N}$ is a learnable matrix transforming N -dimensional features to M -dimensional features, the agg is a permutation-invariant aggregation operation such like element-wise mean-pooling, and the ρ operator can be a non-linear activation function such as ReLU. $\mathcal{N}_{v_i}^{\mathcal{E}}$ is the neighborhood of the node v_i connected by $\mathcal{E}$ in $\mathcal{G}$. In GCN, these two operators are integrated as follows:

$$g_{\mathcal{E}}(v_i) = \text{ReLU} \left(W \cdot \text{Mean} \left(\{ h_{v_j} \mid v_j \in \mathcal{N}_{v_i}^{\mathcal{E}} \cup \{v_i\} \} \right) \right). \quad (3.2)$$

Following on this work, there is increasing interest in extending and improving GCN with more powerful aggregation function such as GraphSAGE [HYL17] and Graph Attention Network (GAT) [VCC17].

Despite advances of GCNs, applying them to bipartite graphs in biomedical domains was seldom explored. The main technical challenge is that node features in different domains of bipartite graphs present quite distinct characteristics. Therefore, it is insufficient to simply apply conventional GCNs to model the connections between multiple domains. To build and explore such connection, we identified that recent studies [HXR19, Nas18] have shown the effectiveness of bipartite graph neural networks on modeling inter-connected graphs.

Conceptually, our research draws inspiration from recent progress of applying GCNs in biomedicine. For example, [ZAL18] achieved state-of-the-art results in predicting polypharmacy side effects by using GCN. [FBS17] showed the effectiveness of GCN in the task of protein interface prediction. [KMB16] proposed a graph convolution framework to learn molecular representations for data-driven tasks considering both node and edge features.

3.4 Datasets

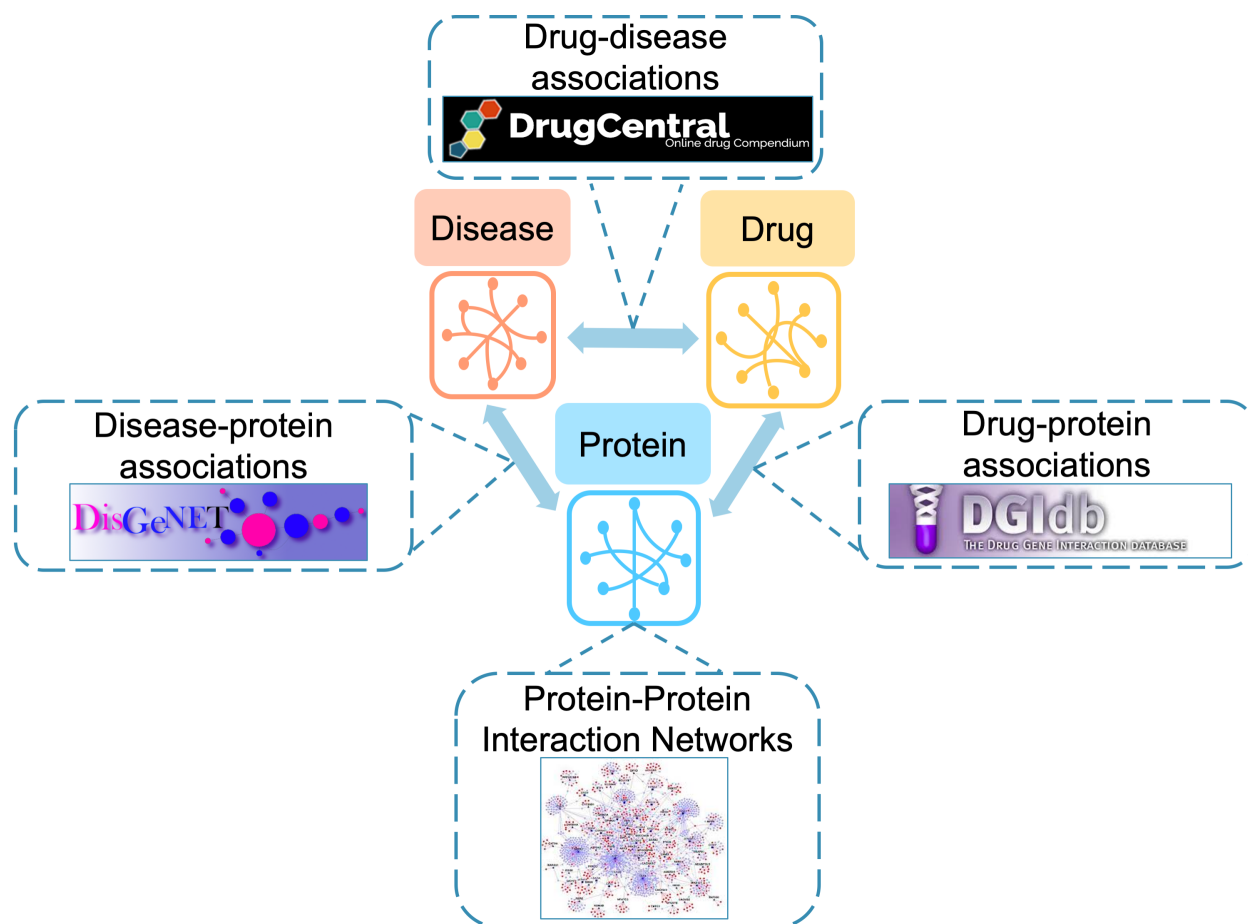


Figure 3.1: Overview of our heterogeneous information network. The multi-relational network has 592 disease, 1,012 drug and 13,460 protein nodes connected by 3,204 drug-disease, 7,713 drug-protein, 104,716 disease-protein and 141,296 protein-protein edges.

We formulated the problem of drug repurposing as a drug-disease link prediction task using multi-modal interaction data. We constructed a multi-relational graph network using multiple biomedical datasets that allow systematic evaluations for drug repurposing. Specifically, drug-disease interaction networks contain drugs therapeutic indications. Drug-protein/disease-protein networks describe the proteins targeted by drugs/diseases. Lastly, protein-protein networks contain interaction relationships between proteins. Below we de-

scribe details of the datasets to construct the graph network in our study.

3.4.1 Drug-Disease Associations

For this study, we collected 3,204 known therapeutic indications of drugs from repoDB database [BP17], in which 6,677 approved indications were drawn from DrugCentral [UHK16]. Only FDA-approved small-molecule drugs were considered and generic name of each drug were standardized by Medical Subject Headings (MeSH) [Lip00] and Unified Medical Language System (UMLS) vocabularies [Bod04]. We also mapped drugs to PubChem [KCC18] with compound ID to get their chemical structure information represented by SMILE string [Wei88]. Most of drugs (75%) treat less than 3 indicated diseases; only 4% of drugs treat more than 10 diseases. 70% Of the diseases have less than 5 drugs; 16% of the diseases have 5-10 drugs; 14% of diseases have more than 10 drugs.

3.4.2 Drug-Protein and Disease-Protein Associations

Drug targets were obtained from the DGIdb [CWF17] database which consolidates drug gene interactions and potentially druggable genes into a single resource from papers, databases and web resources. DGIdb normalized content from 30 disparate sources using a combination of expert curation and text-mining, resulting in 29,783 drug gene interactions which cover 41,100 genes and 9,495 drugs. We pulled target protein-coding genes of a given drug from DGIdb, then mapped genes to proteins with gene names. Our drug-protein interaction network covers 7,713 drug-protein interactions between 1,012 drugs and 1,681 proteins.

Disease-protein associations were extracted from DisGeNET [PBQ16], one of the largest available collections of genes and variants involved in human diseases. DisGeNET integrates data from expert curated databases with information gathered through text-mining the scientific literature, including various resources such as CTD [DGJ16] and OMIM [HSA05]. We pulled protein-coding genes of a given disease and then map them to corresponding

products. Our curated disease-protein interaction network covers 104,716 disease-protein associations between 592 diseases and 9,941 proteins.

3.4.3 Protein-Protein Interactions

We used the human protein-protein interactions (PPI) compiled by [MSK15] which is an unweighted and undirected network with 13,460 proteins and 141,296 physical interactions. The network contains physical interactions with experimental support, such as regulatory interactions, metabolic enzyme-coupled interactions and signaling interactions. The PPI network is approximately scale-free and shows other typical characteristics as observed previously in many other biological networks, such as high clustering and short pathlengths.

The final multi-relational network after linking entity vocabularies across different modalities and databases has 592 disease, 1,012 drug and 13,460 protein nodes connected by 3,204 drug-disease, 7,713 drug-protein, 104,716 disease-protein and 141,296 protein-protein edges.

3.5 Method

3.5.1 Inter-Domain Message Passing through Bipartite Graph Convolution

A bipartite graph $\mathcal{BG}(\mathcal{U}, \mathcal{V}, \mathcal{E})$ is a graph $\mathcal{G}(\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ where $\mathcal{U}$ and $\mathcal{V}$ denote two sets of the two domains of vertices (nodes). u_i and v_j denote the i -th and j -th node in $\mathcal{U}$ and $\mathcal{V}$ respectively, where $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$. All edges of a bipartite graph are strictly between $\mathcal{U}$ and $\mathcal{V}$ (i.e. $\mathcal{E} = \{(u, v) | u \in \mathcal{U}, v \in \mathcal{V}\}$), e_{ij} denotes the edge between u_i and v_j . The features of two sets of nodes can be denoted by X_u and X_v where $X_u \in \mathbb{R}^{M \times P}$ is a feature matrix with $\vec{x}_{u_i} \in \mathbb{R}^P$ representing the feature vector of node u_i , and $X_v \in \mathbb{R}^{N \times Q}$ is defined similarly.

Bipartite graph convolution only performs message passing and node feature aggregation through inter-domain edges as the intra-domain edges are absent in bipartite graphs. For the

message passing $\text{MP}_{v \rightarrow u}$ from domain $\mathcal{V}$ to $\mathcal{U}$, we define a general bipartite graph convolution (bg) as:

$$bg_{\mathcal{E}}(u_i) = \rho \left(\text{agg} \left(\{ W_{u_i, v_j} \vec{x}_{v_j} | v_j \in \mathcal{N}_{u_i}^{\mathcal{E}} \} \right) \right), \quad (3.3)$$

where $\mathcal{N}_{u_i}^{\mathcal{E}}$ is the neighborhood of the node u_i connected by $\mathcal{E}$ in $\mathcal{BG}(\mathcal{U}, \mathcal{V}, \mathcal{E})$ (i.e. $\mathcal{N}_{u_i}^{\mathcal{E}} \subset \mathcal{V}$). Note that any unipartite graph convolution defined on $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be formulated as a bipartite graph convolution defined on $\mathcal{BG}(\mathcal{V}, \mathcal{V}, \mathcal{E})$.

Our bipartite graph convolution layers uses GAT as the backbone, termed as bipartite graph attention convolution layer (bga). As the attention mechanism considers features of two sets of nodes, we specifically define a learnable matrix $W^u \in \mathbb{R}^{P \times S}$ (resp. $W^v \in \mathbb{R}^{Q \times S}$) for X_u (resp. X_v). The bga can be formulated as:

$$bga_{\mathcal{E}}(u_i) = \text{ReLU} \left(\sum_{v_j \in \mathcal{N}_{u_i}^{\mathcal{E}}} \alpha_{u_i, v_j} W^v \vec{x}_{v_j} \right) \quad (3.4)$$

where the weighting coefficients can be expressed as:

$$\alpha_{u_i, v_j} = \frac{\exp \left(\rho \left(\vec{a}^T [W^u \vec{x}_{u_i} \| W^v \vec{x}_{v_j}] \right) \right)}{\sum_{v_k \in \mathcal{N}_{u_i}^{\mathcal{E}}} \exp \left(\rho \left(\vec{a}^T [W^u \vec{x}_{u_i} \| W^v \vec{x}_{v_k}] \right) \right)} \quad (3.5)$$

3.5.2 Model Framework

We cast drug discovery task as a link prediction problem by fusing information from a heterogeneous network incorporating drug, disease and protein relationships. More specifically, we show that the heterogeneous network can be represented by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with three sets of nodes: drugs ($\mathcal{V}^D$), diseases ($\mathcal{V}^S$) and proteins ($\mathcal{V}^P$). The initial features of these three sets of nodes are $X_{\mathcal{V}^D}$, $X_{\mathcal{V}^S}$ and $X_{\mathcal{V}^P}$, respectively. $\mathcal{E}$ consists of three inter-domain edges including drug-disease associations ($\mathcal{E}^{DS}$), drug-protein target relationship ($\mathcal{E}^{DP}$) and disease-protein target relationship ($\mathcal{E}^{SP}$), and one intra-domain edges of PPI network ($\mathcal{E}^{PP}$).

Our model operates directly on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with encoder-decoder architecture (Figure 3.2A). The encoder is a bipartite graph convolutional network learning the embedding

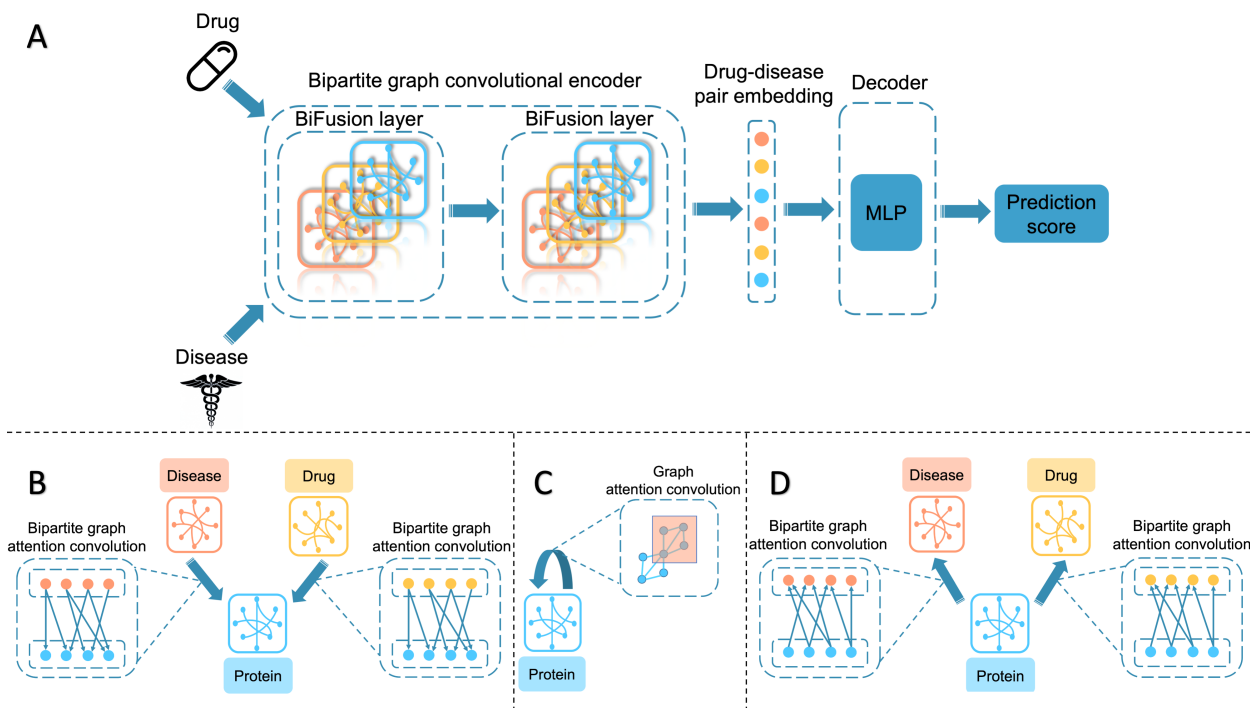


Figure 3.2: Overview of BiFusion model architecture. (A) The pipeline of BiFusion contains a bipartite graph convolutional network encoder and a MLP decoder. The encoder takes similarity features of drug and disease nodes as inputs, and generates drug-disease pair embeddings by fusing heterogeneous information through message passing across drug, disease and protein nodes. Each BiFusion layer consists of three computing steps shown in the following sub-figures. BiFusion decoder takes pair embeddings to produce prediction score and reconstruct drug-disease association matrix. (B) The first step in BiFusion layer: a single bipartite graph attention convolution layer is applied to project information from drug and disease domains to protein domain. (C) The second step in BiFusion layer: a single layer graph attention convolution layer is applied within PPI network. (D) The third step in BiFusion layer: another bipartite graph attention convolution layer is used to update drug and disease features based on learnt protein node embeddings.

representations for all graph nodes. It fuses heterogeneous information through message passing across drug, disease and protein nodes. The decoder is a multi-layer perceptron

neural network decoder using drug and disease node embeddings to reconstruct drug-disease association matrix.

3.5.3 Node Feature Representation

We applied zero-initialization for all protein nodes and defined similarity measures for initializing features for drug and disease nodes.

- **Chemical-based Drug Similarity Measure:** Canonical SMILES of the drug molecules were used from PubChem. The similarity score between two drugs is computed based on their fingerprints according to the two-dimensional Tanimoto score [Tan57].
- **Graph-based Disease Similarity Measure:** We used Mesh term [Lip00] as disease descriptor for constructing similarity measures. Given that the structure of MeSH is a directed acyclic graph which enables the comparison of semantic similarity in the graph, we applied the graph-based method proposed by [WDP07] to measure similarity between disease MeSH terms.

3.5.4 Bipartite Graph Convolutional Encoder

Each layer of our bipartite graph convolutional encoder consists of three computing steps. First, we applied a single bipartite graph attention convolution layer to pass the message of drugs and diseases to target proteins simultaneously. Conceptually, we can view this step as projecting information from macro level (e.g. information in drugs and disease domains) to micro level (e.g. protein space). This message passing step is formulated as follows:

$$\begin{cases} \text{MP}_{\mathcal{V}^D \rightarrow \mathcal{V}^P}^{(k)} : h_{v_i^p}^{(k)1} = \text{bga}_{\mathcal{E}^{DP}}(v_i^p) \\ \text{MP}_{\mathcal{V}^S \rightarrow \mathcal{V}^P}^{(k)} : h_{v_i^p}^{(k)2} = \text{bga}_{\mathcal{E}^{SP}}(v_i^p) \end{cases} \quad (3.6)$$

where k indicates the layer index and $h_{v_i^p}^{(k)}$ are hidden embeddings of nodes v_i^p (when $k = 0$, $h_{v_i^p}^{(0)} = x_{v_i^p}$). We concatenated the results of two message passing processes into a unified embedding representation. Therefore, the updated embeddings of protein nodes can be written as:

$$H_{\mathcal{V}^P}^{(k)} = H_{\mathcal{V}^P}^{(k)1} \parallel H_{\mathcal{V}^P}^{(k)2} \quad (3.7)$$

In the second step, to enhance domain fusion and model the relationships between drug targets and disease-gene products, we applied a single layer GAT within our PPI network. The intuition behind this step is that GAT can enable feature smoothing between protein neighborhood nodes, and a drug is more likely to treat a disease if they are nearby in protein space. This layer performs intra-domain message passing that allows information fusion in protein space and depict complex interactions between drugs and diseases. Therefore, protein nodes serve as a bridge of message passing within our multi-relational graph. In each layer, GAT propagates node hidden embeddings across edges of PPI network, which is defined as:

$$\text{MP}_{\mathcal{V}^P \rightarrow \mathcal{V}^P}^{(k)} : h_{v_i^p}^{(k)} = \text{bga}_{\mathcal{E}^{PP}}(v_i^p) \quad (3.8)$$

Finally, we utilized the non-linear graph information captured by protein nodes to update hidden embeddings of drug and disease nodes. In particular, we applied another bipartite graph attention convolution layer to project protein embeddings back to drug and disease domains. Therefore, the third step can be viewed as an integrative graph method to learn drug and disease representations through closeness in PPI network. For those drugs and diseases that share target proteins, this step will help to further make their features similar. The updated feature representations of drug and disease nodes can be written as:

$$\begin{cases} \text{MP}_{\mathcal{V}^P \rightarrow \mathcal{V}^D}^{(k)} : h_{v_i^d}^{(k)} = \text{bga}_{\mathcal{E}^{DP}}(v_i^d) \\ \text{MP}_{\mathcal{V}^P \rightarrow \mathcal{V}^S}^{(k)} : h_{v_i^s}^{(k)} = \text{bga}_{\mathcal{E}^{SP}}(v_i^s) \end{cases} \quad (3.9)$$

We summarize our bipartite graph convolutional encoder in Figure 3.2B, C and D.

3.5.5 Multi-layer Perceptron Neural Network Decoder

Our network decoder applies a multi-layer perceptron to reconstruct links in the drug-disease interaction graph. In particular, using embeddings of drug node $h_{v_i^d}$ and disease node $h_{v_j^s}$ returned by the encoder, we concatenated two embeddings to represent the drug-disease pair and then fed into the decoder. In particular, decoder scores a drug-disease pair (v_i^d, v_j^s) through a three-layer MLP neural network representing how likely it is that the drug can be indicated for the disease:

$$\text{MLP}(v_i^d, v_j^s) = \text{MLP}(h_{v_i^d} \parallel h_{v_j^s}) \quad (3.10)$$

Then we applied a sigmoid function σ to compute probability of edge (v_i, v_j) :

$$p^{ij} = \sigma(\text{MLP}(v_i^d, v_j^s)) \quad (3.11)$$

3.5.6 Model Training

During model training, we optimized model parameters using the cross-entropy loss in an end-to-end fashion. Followed previous studies [MSC13], we trained the model through negative sampling. Specifically, for each positive drug-disease edge (v_i, v_j) , we sampled a random edge (v_i, v_k) as a negative example. This is achieved by replacing a drug or disease node v_j with node v_k that is selected randomly according to a sampling distribution P_s [MSC13]. We calculated the final loss function by considering all edges. To optimize the model, we used the Adam optimizer [KB14] and initialized weights as described in [GB10]. In order to generalize well to unobserved data, we trained the model in a denoising setup by randomly dropping out all outgoing messages of a particular node with a fixed probability. In particular, during the message passing process in encoder, individual outgoing messages across multi-relational edges are dropped out independently, making embeddings more robust against the presence or absence of single edges. We also apply regular dropout [SHK14] to the hidden layer units in MLP decoder.

3.6 Experiments

3.6.1 Evaluation Metrics

We conduct ten-fold cross validation to evaluate the model performance. All known drug-disease associations are randomly divided into ten subsets with equal size. A matching number of unknown pairs were selected as negative samples in training and testing sets through negative sampling strategy. In each cross validation trial, one subset is taken in turn as the test set, while the remaining subsets constitute the training set. We selected model hyperparameters by performing cross-validation on the training set. We measure the prediction performance using three criteria: area under the receiver-operating characteristic (AUROC), area under the precision-recall curve (AUPRC) and overall accuracy, which are widely used for drug indication prediction tasks. Since the prediction performance can vary considerably across diseases and drugs, we further report the disease/drug-centric accuracy, which is the average of balanced accuracy of all drug-disease pair subsets clustered by disease/drug nodes. To reduce the data bias, we performed 100 independent cross-validation runs and reported the full distribution of average testing performance of all evaluation metrics. Additionally, during each random run, a different sampled negative set and partition of the dataset were used. To prevent information leakage in the evaluation, we ensure that only drugs and diseases seen in the training set were used to construct similarity features.

3.6.2 Method Comparison

We compare the performance of our model against the several competing approaches. Specifically, BiFusion uses a 2-layer architecture with 256 and 128 hidden units in each layer, and a dropout rate of 0.1 in all experiments.

- **GCN** [ZAL18] includes encoder and decoder modules. The encoder is a conventional GCN operating on our multi-modal graph of protein-protein, drug-protein and disease-

protein interactions. The decoder is a tensor factorization model using node embeddings to model drug-disease associations.

- **DeepWalk** [PAS14] learns latent node representations of our heterogeneous information network based on local information obtained from truncated random walks. Drug-disease pairs are represented by concatenating latent drug and disease node representations. We used pair representations as inputs to train a logistic regression classifier.
- **Collective variational autoencoder (cVAE)** [CR18, ZZL19] simultaneously recovers drug-disease association matrix and side information by using a variational autoencoder. Specifically, drug-disease association matrix and drug-drug similarity matrix are encoded and decoded collectively through the same inference network and generation network.
- **Sparse Linear Methods with Side information (SSLIM)** [NK12] learns a sparse coefficient matrix to do top-N recommendation, by leveraging both association matrix and similarity matrix within a regularized optimization process.
- **Network-based proximity:** This approach measures relative proximity that quantifies the network-based relationship between drugs and disease proteins in the interactome. Given the set of disease proteins S and drug targets T , the proximity is the closest measure $d(s, t)$, the shortest path length between nodes s and t in the network, which is defined $d(S, T) = \frac{1}{\|T\|} \sum_{t \in T} \min_{s \in S} d(s, t)$. Proximity versus sensitivity and specificity curves is used to find the optimal proximity threshold.
- **BiFusion-v2 (w/o PPI):** To investigate the contribution of intra-domain message passing operation in PPI network, we implemented a variant of our BiFusion model. We remove the second step of message passing to hide PPI information in each layer of encoder.

3.7 Results

3.7.1 Performance Comparison

Table 3.1: The summary of model performance on repoDB dataset under ten-fold cross validation.

Method	AUROC	AUPRC	Overall Accuracy	Drug-centric Accuracy	Disease-centric Accuracy
BiFusion	0.857 ± 0.003	0.867 ± 0.003	0.738 ± 0.002	0.710 ± 0.003	0.705 ± 0.003
BiFusion-v2 (w/o PPI)	0.837 ± 0.003	0.810 ± 0.003	0.712 ± 0.003	0.687 ± 0.002	0.674 ± 0.003
GCN	0.792 ± 0.004	0.774 ± 0.005	0.700 ± 0.003	0.651 ± 0.004	0.659 ± 0.004
DeepWalk	0.769 ± 0.003	0.764 ± 0.003	0.672 ± 0.003	0.617 ± 0.003	0.637 ± 0.003
cVAE	0.743 ± 0.003	0.739 ± 0.003	0.665 ± 0.002	0.623 ± 0.003	0.616 ± 0.003
SSLM	0.701 ± 0.002	0.703 ± 0.002	0.635 ± 0.002	0.590 ± 0.002	0.625 ± 0.002
Network-based proximity	0.663 ± 0.004	0.678 ± 0.004	0.608 ± 0.004	0.568 ± 0.004	0.603 ± 0.005

In Table. 3.1, we found that BiFusion showed strong performance and outperformed other approaches by a large margin. Especially, BiFusion surpassed other methods without incorporating the graph structure information (cVAE and SSLM) by up to 22.3% (AUROC), highlighting the importance of graph-level information fusion for drug-disease findings. We also observed that our model achieved a gain of 8.2% (AUROC) over GCN operating on homogeneous graph, which indicated the effectiveness of bipartite graph convolution to model multi-relational network. Additionally, the protein feature smoothing operation allowed BiFusion a 2.4% gain (AUROC) over BiFusion-v2. Such finding supported that intra-domain message passing operation can encourage information fusion in PPI network and thus enhanced the model performance on capturing complex interactions between drugs and diseases.

3.7.2 Investigation of Novel Predictions

To validate the ability of models for predicting truly novel drug-disease associations (i.e. for new diseases without any treatment information), we further implemented a disjoint cross

Table 3.2: The results of novel predictions on repoDB dataset.

Method	AUROC	AUPRC	Overall Accuracy	Drug-centric Accuracy	Disease-centric Accuracy
BiFusion	0.775 $\pm$ 0.003	0.794 $\pm$ 0.003	0.709 $\pm$ 0.002	0.666 $\pm$ 0.003	0.700 $\pm$ 0.003
BiFusion-v2 (w/o PPI)	0.749 $\pm$ 0.003	0.732 $\pm$ 0.003	0.674 $\pm$ 0.003	0.663 $\pm$ 0.003	0.668 $\pm$ 0.004
GCN	0.740 $\pm$ 0.004	0.726 $\pm$ 0.005	0.687 $\pm$ 0.004	0.669 $\pm$ 0.004	0.656 $\pm$ 0.005
DeepWalk	0.712 $\pm$ 0.004	0.700 $\pm$ 0.004	0.663 $\pm$ 0.003	0.647 $\pm$ 0.003	0.655 $\pm$ 0.004
cVAE	0.696 $\pm$ 0.003	0.698 $\pm$ 0.003	0.637 $\pm$ 0.002	0.631 $\pm$ 0.002	0.641 $\pm$ 0.003
SSLIM	0.671 $\pm$ 0.002	0.699 $\pm$ 0.003	0.616 $\pm$ 0.002	0.575 $\pm$ 0.003	0.591 $\pm$ 0.002
Network-based proximity	0.661 $\pm$ 0.004	0.692 $\pm$ 0.004	0.622 $\pm$ 0.004	0.574 $\pm$ 0.004	0.594 $\pm$ 0.005

validation fold generation method (disease-centric cross validation) that ensures none of the diseases in one fold would appear in another fold. Specifically, all disease nodes were split into 10 equal-sized subsets during disease-centric cross validation. We clustered drug-disease pairs by disease nodes, then recombined pair clusters based on disease subsets resulting in 10 pair subsets. Each pair subset was used in turn as the testing set. We also performed 100 independent runs to report full distribution of average testing performance.

As shown in Table.3.2, BiFusion achieved AUROC value of 0.775 and disease-centric accuracy value of 0.700, outperforming all baseline methods. We observe that BiFusion surpassed two other GCN-based methods by up to 6.7% (disease-centric accuracy), which showed the superior performance of our model on predicting novel drug-disease associations.

3.7.3 Experiments on the External Dataset

Table 3.3: The summary of model performance on external dataset.

Method	AUROC	AUPRC	Overall Accuracy	Drug-centric Accuracy	Disease-centric Accuracy
BiFusion	0.757 $\pm$ 0.005	0.721 $\pm$ 0.004	0.671 $\pm$ 0.004	0.675 $\pm$ 0.004	0.653 $\pm$ 0.004
BiFusion-v2 (w/o PPI)	0.722 $\pm$ 0.005	0.677 $\pm$ 0.005	0.675 $\pm$ 0.005	0.670 $\pm$ 0.004	0.636 $\pm$ 0.005
GCN	0.717 $\pm$ 0.004	0.676 $\pm$ 0.004	0.664 $\pm$ 0.003	0.667 $\pm$ 0.003	0.624 $\pm$ 0.004
DeepWalk	0.649 $\pm$ 0.003	0.628 $\pm$ 0.003	0.611 $\pm$ 0.003	0.604 $\pm$ 0.003	0.572 $\pm$ 0.003
cVAE	0.676 $\pm$ 0.006	0.653 $\pm$ 0.005	0.637 $\pm$ 0.005	0.629 $\pm$ 0.006	0.639 $\pm$ 0.006
SSLIM	0.652 $\pm$ 0.003	0.607 $\pm$ 0.003	0.602 $\pm$ 0.002	0.614 $\pm$ 0.003	0.625 $\pm$ 0.003
Network-based proximity	0.610 $\pm$ 0.003	0.579 $\pm$ 0.002	0.573 $\pm$ 0.003	0.566 $\pm$ 0.002	0.563 $\pm$ 0.003

In order to illustrate the potential generalization of our model, we performed evaluation on an external dataset [GSR11]. Following the rule of collecting our primary dataset, we identified a total of 1,234 associations containing 475 drugs and 141 diseases from the external dataset. We compared the performance of our method with baseline approaches under the same experiment settings as discussed. Table 3.3 showed the full distribution of average testing performance of 100 random runs. The results illustrated that BiFusion led best performance with AUROC value of 0.757, while GCN, DeepWalk, cVAE and SSLM have 0.717, 0.649, 0.676 and 0.652 respectively. BiFusion also achieved the best result in AUPRC with surpassing baseline methods by up to 24.5%.

3.7.4 Case Study

Table 3.4: New candidate drugs ranked by prediction scores by BiFusion for breast carcinoma and Parkinson disease.

Diseases	Rank	Candidate drugs	Evidences
Breast carcinoma	1	Clofarabine	[LKS14, LKC18]
	3	Cimetidine	[BKT03]
	4	Thiamine	[LMU18]
	5	Arsenic trioxide	[ZMP16, SCH17a]
Parkinson disease	1	Dextromethorphan	[FMN17, FSK19]
	2	Solifenacin	[ZEV15]
	4	Atomoxetine	[WOB18, RNR16, YAN15]
	7	Venlafaxine	[BLK16]
	8	Tapentadol	[VSC20]

We conducted a case study to further assess the quality of our model’s novel predictions by performing a literature-based evaluation of new hits. Specifically, we applied BiFusion to predict candidate drugs for two diseases including breast carcinoma and Parkinson’s disease.

After the prediction scores of all candidate pairs are computed, we generate a ranked list of drug-disease associations by the predicted scores. We then identify novel associations by excluding all the known drug-disease associations from the primary dataset. Table 3.4 shows candidate drugs with evidences.

- **Breast carcinoma:** Among the top 5 predicted drugs in the rank list, 4 drugs (80% success rate) were validated by various literature evidences. Arsenic trioxide was predicted by BiFusion to be associated with breast carcinoma, which is supported by recent reports. For example, [ZMP16] showed Arsenic trioxide suppresses cell growth and migration via inhibition of miR-27a in breast cancer cells. And [SCH17a] found that Arsenic trioxide suppressed cell growth, stimulated apoptosis, and retarded cell invasion partly via upregulation of let-7a in breast cancer cells. Clofarabine (CIF) is also one of the top predicted candidate for treating breast carcinoma. [LKS14] provided the first evidence of CIF implications in epigenetic regulation of transcriptional activity of selected tumour suppressor genes in breast cancer. [LKC18] demonstrated the ability of CIF-based combinations with polyphenols to promote cancer cell death and reactivate DNA methylation-silenced tumor suppressor genes in breast cancer cells. In addition, BiFusion found that Cimetidine and Thiamine were associated with breast cancer, which was supported by several evidences [BKT03, LMU18].
- **Parkinson’s disease:** Parkinson’s disease (PD) is a neurodegenerative disease currently without efficacious treatments available yet. Among top 10 predicted candidates, we found 5 drugs were validated by literature. For example, Dextromethorphan is the top predicted candidate. Despite approval by the FDA for pseudobulbar affect based on studies of patients with amyotrophic lateral sclerosis (ALS) or multiple sclerosis (MS), [FMN17, FSK19] both provide evidence of clinical benefit with dextromethorphan-quinidine for treating PD. Atomoxetine was also predicted by our model to be associated with PD. Such prediction can be supported by a previous study [YAN15] indicating that atomoxetine can enhanced prefrontal cortical activation and

frontostriatal connectivity and may improve response inhibition in Parkinson’s disease. The results of [RNR16] also suggested that atomoxetine restores the response inhibition network in Parkinson’s disease.

3.7.5 The Effect of Layer Numbers on Model Performance

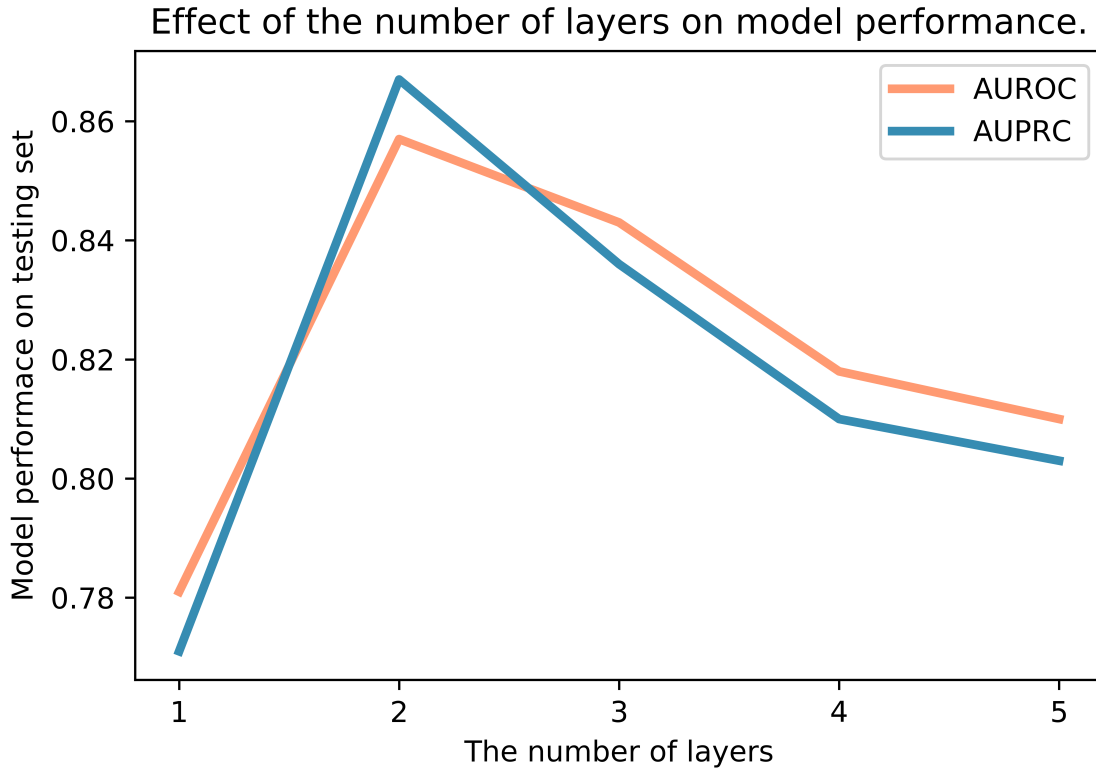


Figure 3.3: Effect of the number of layers on model performance. The x axis denotes the number of BiFusion layers and the y axis is the model performance on testing set.

To investigate the effect of layer numbers on model performance, we compared results with different number of layers in BiFusion on the repoDB dataset. We performed 100 independent cross-validation runs and reported the mean value of AUROC and AUPRC. Figure 3.3 showed the model performance along with the increase of layer numbers. We observed that one layer has the lowest performance, suggesting that a shallow bipartite GCN

can not sufficiently propagate the node feature to fuse heterogeneous information, especially for the complex drug-protein-disease network. Meanwhile, we found that BiFusion achieved significant improvement with two layers’ structure. But with more than two layers, the model performance tends to decrease. This finding may be explained that GCN model is viewed as a special form of Laplacian smoothing that over-smoothing occurs with too many convolutional layers [LHW18]. Thus if BiFusion’s layers are going too deep, the output embedding features can be over-smoothed and less differentiated from different classes.

3.8 Conclusion

In this study, we presented a novel bipartite graph convolutional networks towards heterogeneous information fusion for computational drug repurposing. Our BiFusion model achieved information fusion via an important inter-domain message passing across drug-, disease- and protein-level information. Extensive experiments have demonstrated that our model achieves strong performance on the task of drug repurposing. In addition, we externally validated results that have confirmed the potential generalization of our approach for drug repurposing. Case study offers concrete examples that reaffirmed medical usefulness of our approach. In the future work, we plan to assess model performance by exploring scalable cohorts with clinically-validated associations between drugs and diseases. Since our approach supports multi-level biological information fusion, additional pharmaceutical information such as drug-side effects information can be also considered to improve our network analysis.

CHAPTER 4

Temporal Graph Learning using Heterogeneous Electronic Health Records to Predict Depression

4.1 Abstract

Advances in deep learning algorithms have significantly improved disease modeling with electronic health record (EHR) data. To address the challenge of early depression diagnosis for timely intervention, we propose a temporal graph learning approach that integrates heterogeneous EHR information. Our method reconstructs sequential EHR data into temporal graphs, combining time-aware information propagation with a heterogeneous graph attention mechanism. This allows us to capture dynamic patterns by constructing temporal graphs connecting different patient record sequences. Additionally, our framework includes a graph model explanation component, providing clearer insights into learned patterns. Tested on a large-scale EHR dataset from the UCLA Health system, our algorithm effectively uncovers relationships between patient medical status and depression diagnosis. Through rigorous evaluation against transformer and graph-based baselines, our model consistently outperforms benchmarks across various metrics and time windows. Its robust performance underscores its suitability for real-world clinical applications, offering both technical contributions and the potential to enhance accuracy and interpretability in depression prediction. This advancement holds promise for improving clinical decision support systems for timely and personalized patient care.

4.2 Introduction

Depression, as the most prevalent mental health disorder globally, exhibits a wide-ranging one-year prevalence rate of 7 to 21% [LTL18]. Its profound impact on public health is underscored by its association with a significant number of annual suicides [CGF14]. Particularly concerning is its frequent co-occurrence with chronic diseases, amplifying the risks of self-harm, premature mortality, and the emergence of comorbid conditions such as heart disease, stroke, and obesity [MHY06, MRO18]. Patients experiencing depressive symptoms are considerably more susceptible to developing major depressive disorder within a year [MRO18]. Despite its prevalence and the substantial economic burden it imposes, the effectiveness of current clinical screening methods for high-risk depression patients is limited, achieving only a 50% true positive rate. This has catalyzed significant informatics efforts aimed at improving diagnostic accuracy through the deployment of machine learning algorithms [MVR09, HLI14, Jin15, ZXH15].

Diagnosing depression poses a significant challenge, particularly for non-specialist clinicians who often depend on frameworks like the World Health Organization’s International Classification of Diseases, Version 10 (ICD-10). Complementing these are patient self-reports collected through standardized questionnaires such as the Patient Health Questionnaire-9 (PHQ-9), which is valued for its reliability in screening but is recommended primarily for initial assessments rather than as a standalone diagnostic tool [SK11, Kro21]. Research, including a meta-analysis by [DM19], highlights that early interventions can reduce the incidence of depression by about 20% within 3 to 24 months. Delayed diagnoses, however, may exacerbate patient suffering and escalate resource use for both individuals and healthcare systems [MV13]. At the same time, the fusion of electronic health records (EHRs) with machine learning has greatly enhanced early diagnostics in mental health care. Over the past decade, EHR systems have transitioned from simply boosting healthcare efficiency to integrating comprehensive patient medical histories, combining structured data like diagnoses

and lab tests with unstructured data such as clinical notes and medical images.

In contrast to static datasets, EHR data offer longitudinal insights into patients’ medical journeys, enabling the discovery of latent patterns. However, the intricate nature and irregular distribution of clinical events within temporal EHR data present substantial technical hurdles, including irregularity, sparsity, heterogeneity, and model opacity [XYN22]. Addressing these challenges, we introduce a temporal graph learning method that constructs large-scale graphs to dynamically model the evolving interactions and dependencies among EHR sequences. Our method employs time-aware information propagation and a heterogeneous attention mechanism, significantly enhancing the model’s ability to identify patients at an elevated risk of depression within specific time windows. By integrating these components, our approach not only improves the accuracy and explainability of depression predictions but also boosts efficacy and utility in real-world clinical settings for proactive depression screening and early intervention.

4.3 Related Work

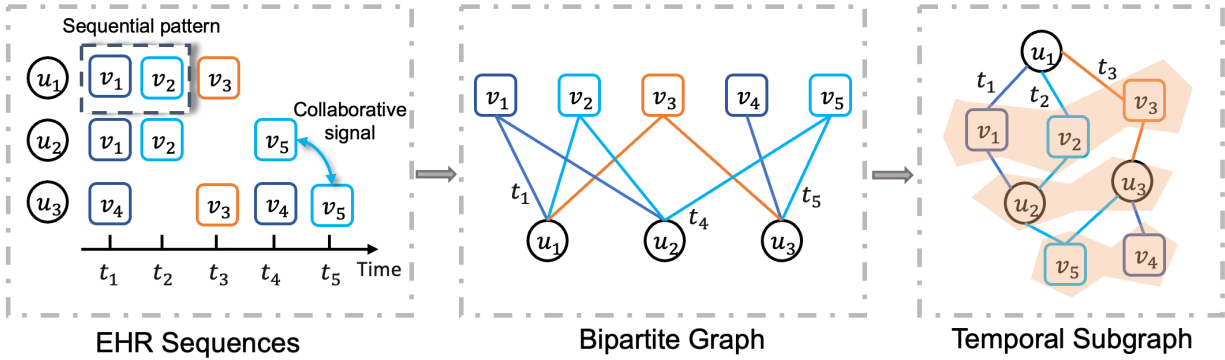


Figure 4.1: Illustration of transforming EHR sequences into temporal graphs. Our approach samples temporal subgraphs from patient-code bipartite graphs to more effectively capture high-order collaborative signals among patient sequences.

4.3.1 Temporal Deep Learning for EHR

Temporal deep learning methods have emerged as a promising avenue for leveraging the longitudinal nature of EHR data, which are often documented with timestamps to track patients’ conditions over time. Recognizing the importance of both long- and short-term trends within EHR trajectories for various prediction tasks, researchers have primarily employed recurrent neural networks (RNNs) and attention mechanisms to model longitudinal dependencies [AOE23]. While RNNs, such as bidirectional GRU and LSTM, have demonstrated success in preserving information across sequential steps [RGS21, MSO20, ROC18], self-attention mechanisms have been utilized to determine the significance of individual visits and the relationships among different entities across visits [VSP17, LYX20, MCZ17, MSO21].

However, despite the potential of sequential deep learning architectures in capturing the temporal dynamics of EHR data, they often overlook crucial aspects of temporal contextual information. Specifically, these methods tend to treat all patient EHR trajectories as equally significant, modeling them separately without considering potential collaborative signals among different sequences. This assumption may not hold true in real-world scenarios, leading to limitations in capturing the intricate dynamics of complex medical scenarios. Moreover, existing models frequently fail to simultaneously account for the temporal dynamics of both patients and EHR codes, thereby overlooking high-order collaborative information between them. As a result, these models are limited in their ability to effectively leverage dependencies among different patient histories within relatively long sequences, hindering their capacity to capture the nuanced dynamics inherent in diverse clinical contexts.

4.3.2 Temporal Graph Neural Network

Expanding from sequential models to graph-based frameworks, recent research has explored the integration of graphical structures underlying EHR data to enhance prediction tasks [CXL20]. While these initial efforts primarily focused on analyzing single patient encounters,

lacking the incorporation of temporal dynamics and sequential learning across patient visit trajectories. To address this limitation, we propose the utilization of temporal graphs to encapsulate the evolving dependencies among interconnected patient EHR sequences.

Temporal graph representation offers a natural framework for capturing high-order temporal contextual information within complex systems that evolve over time [SGM21]. The advent of graph neural networks (GNNs) has further propelled advancements in graph learning, enabling the effective modeling of graph dependencies through recursive information propagation and aggregation mechanisms. In the context of temporal graph modeling, Temporal Graph Neural Networks (TGNNs) have emerged as a promising approach, combining deep time series encoding with message passing strategies. These methods encompass both discrete- and continuous-time approaches, with discrete-time methods typically integrating GNNs with time series models such as RNNs and LSTM networks [MRM20, SWG20, WCC20]. Moreover, integrative approaches have been proposed to incorporate spatial and temporal information within a single layer, enhancing model efficiency and effectiveness [CPD21, LHC19, JJQ19]. In contrast, continuous-time approaches focus on refining graph models to capture fine-grained temporal information, utilizing techniques such as temporal point processes parameterized by neural networks and novel time embedding methods [TFB19, KAT19, HJW19, KGE19, XRK19]. Notably, recent advancements [XRK20, RCF20] have further extended the capabilities of TGNNs by incorporating continuous-time embeddings and memory modules to capture evolving node histories. Despite these advancements, it is important to note that most GNNs are designed for homogeneous graphs, which may pose challenges in accurately representing the heterogeneous structures inherent in real-world EHR data.

4.4 Problem Formulation

4.4.1 Temporal Heterogeneous Graph

A heterogeneous graph is characterized by a directed graph $G = (V, E, A, R)$, in which each node $v \in V$ and each edge $e \in E$ are linked with their respective type mapping functions $\tau(v) : V \rightarrow A$ and $\varphi(e) : E \rightarrow R$. For any given edge $e = (v_i, v_j)$ that connects a source node v_i to a destination node v_j , its edge relation is expressed as $\langle \tau(v_i), \varphi(e), \tau(v_j) \rangle$. And naturally $\varphi(e)^{-1}$ represents the inverse of $\varphi(e)$. To accurately reflect the temporal nature of real-world heterogeneous graphs, we define an edge $e = (v_i, v_j, t_k)$ as representing a connection established by node v_i with node v_j at time t_k . If a node v_i interacts with a node v_j on multiple occasions, this would result in several edges, each marked with distinct timestamps to reflect the different times at which connections were established. This methodology underlines that the timestamp associated with any given edge is fixed, indicating the precise time at which it was formed.

4.4.2 Graph Representation of EHR

We formulate the task of predicting depression as classifying future nodes by learning the temporal node embeddings of patients and EHR codes in a continuous-time temporal graph. We denote the set of patients as $\mathcal{U}$ and the set of EHR codes as $\mathcal{V}$. To account for the heterogeneity of EHR code sources, we divide code nodes into three categories: diagnosis, medication, and procedure codes. $S^u = (v_1, v_2, \dots, v_{n_u})$ is the chronological interaction sequence of patient u , where n_u denotes the length of the sequence. $T^u = (t_1, t_2, \dots, t_{n_u})$ is the corresponding timestamp sequence of S^u . All patient-code interactions can be converted into an bipartite temporal graph $\mathcal{G}$ with two sets of nodes. When a patient u interacts with a code v at time t , this interaction is represented by a directed edge $e = (u, v, t)$ from u to v . Depending on the type of code node $\tau(v)$, we assign a unique edge relation $\varphi(e)$, and include a reverse edge with $\varphi(e)^{-1}$ going back from v to u . For a patient u , at any time t before the

depression diagnosis within specified prediction windows, we can capture a snapshot $\mathcal{G}_u^t$ of the graph $\mathcal{G}$ by omitting interactions involving u after time t . Given a patient’s interaction sequence S^u and the associated timestamps T^u , we cast the task of predicting depression as predicting the future node category of patient node u in $\mathcal{G}_u^{t_{n_u}}$.

4.5 Data Preprocessing

4.5.1 Sequential Dataset Construction

We constructed comprehensive datasets from over two decades of EHRs from UCLA hospitals to predict depression. Our work began with identifying over 440 thousand primary care patients from databases. The raw EHR data, initially in SQL table format, were transformed into pandas data frames for analysis in Python.

4.5.1.1 Data Standardization and Reduction

We treated each patient visit as a sequence comprising diagnosis, procedure, medication codes, and demographic information. Diagnosis codes were updated to the ICD-10, converting any ICD-9 codes using General Equivalence Mappings for consistency. Procedure codes were standardized to the Current Procedural Terminology (CPT) format, and medications were coded using the Generic Product Identifier (GPI) system. Demographics were limited to the patient’s gender and age. Given the high dimensionality and sparsity of full ICD-10 codes in our dataset, we reduced codes’ granularity by grouping diagnosis codes according to the three digits preceding the decimal point. Similarly, we simplified GPI codes by removing the last four digits, which indicate dosage form and strength. We further excluded rare codes appearing in less than 1% of our patient database. This process resulted in 1,377 unique diagnosis codes, 1,086 medication codes, and 1,678 procedure codes.

4.5.1.2 Data Labeling

Our dataset comprises patients identified retrospectively, who were not initially suspected of depression. Consequently, conventional methods for depression assessment, such as the Patient Health Questionnaire (PHQ-9) scores, were unavailable. To address this, we identified patients with depression through the presence of depression-related codes in their EHR sequences. The earliest occurrence of any such code was designated as the depression diagnosis time. To accommodate the heterogeneity of depression diagnosis and maintain the integrity of our data labeling process, we differentiated depression codes into two categories: 1) "Clean" depression codes, which are specific ICD codes that directly indicate a depression diagnosis, and 2) "Noisy" depression codes, identified by blending prior medical knowledge and analyzing odds ratios across all codes. High odds ratio codes, suggesting a strong likelihood of indicating depression, were considered noisy. These include codes for conditions such as acute anxiety, as well as codes for anti-anxiety medications and psychotherapy procedures.

Implementing this methodology, we constructed two positive patient groups: one comprising patients with only "clean" depression codes, ensuring a clear diagnosis of depression, and another with patients characterized by "noisy" codes, who might not have explicit depression diagnosis codes due to incomplete EHR histories or misdiagnoses. For individuals without depression, we replaced their diagnosis time with the last EHR timestamp. Considering the predominance of non-depressed cases in our dataset, we selectively retained those without any depression-related codes to enhance the quality of our control group in both the clean and noisy datasets. This meticulous data labeling not only sharpens our depression prediction model's focus but also broadens its applicability across diverse real-world scenarios, facilitating rigorous testing of its generalization capabilities.

4.5.1.3 Time Window Definition

To predict the future risk of depression in a prospective manner, we established varied time windows for patient EHR analysis, similar to the previous works [MSO20, MSO21]. We defined three key decision points preceding a depression diagnosis: three months, six months, and one year, with each data window spanning five years to capture comprehensive historical information. This methodology ensured the exclusion of all depression related codes from patient sequences, eliminating the risk of data leakage. To assess the impact of temporal dynamics and dataset size on model performance, we maintained a consistent patient count across the three predictive scenarios. This was achieved by selecting only those patients who had at least one code present in all three timeframes.

4.5.2 Graph Dataset Construction

We transformed the processed sequential dataset into temporal graphs following the methodology outlined in Section. 4.4. These graphs are composed of patient and code nodes, where connections exist exclusively between these two node types. The code nodes are further categorized into diagnosis, medication, and procedure codes. Depending on the code node category, we assigned distinct edge types to connections originating from patient nodes to code nodes and vice versa. Each edge carries a timestamp derived from the raw EHR dataset, indicating the date of the recorded event. To simplify the graph’s complexity, we aggregated timestamps from daily to quarterly intervals, consolidating edges occurring within the same quarter and involving the same pair of nodes. Through this process, we compiled a clean (noisy) graph dataset comprising over 106.5 (112.4) thousand patients with a 9.7% (14.5%) positive rate, 10.1 (10.5) million diagnosis edges, 2.7 (2.8) million medication edges, and 3.2 (3.3) million procedure edges.

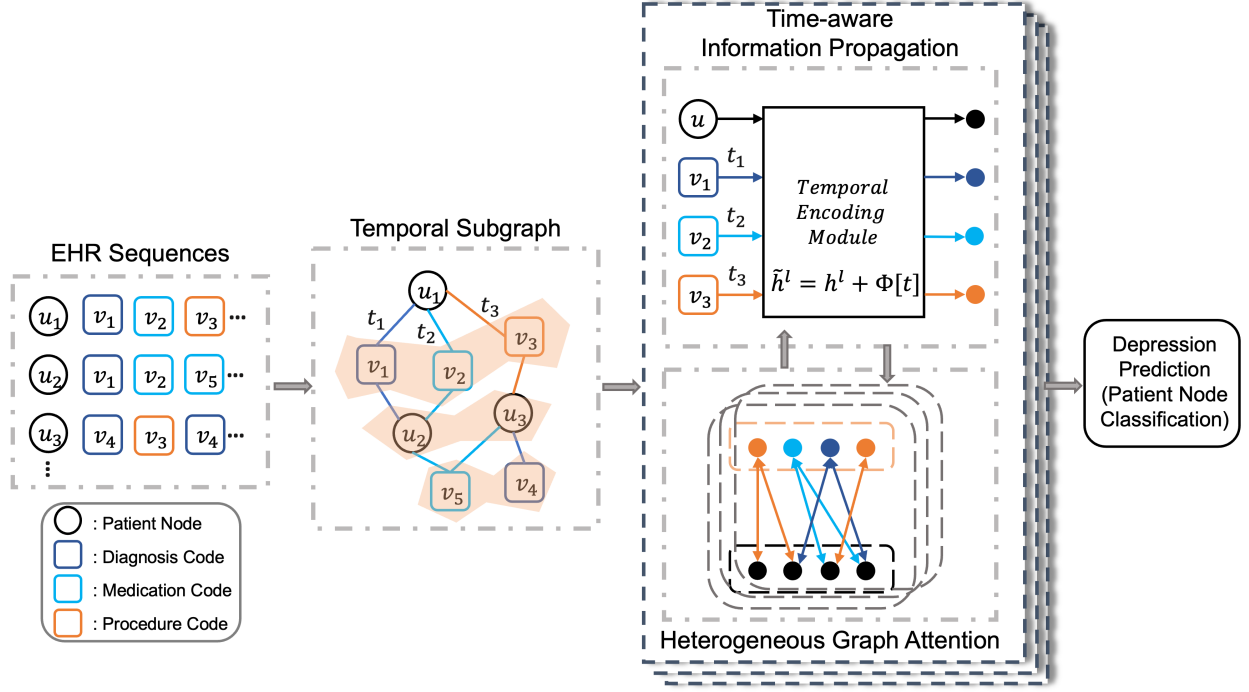


Figure 4.2: Overview of proposed model architecture. Temporal encoding module quantifies temporal effects using edge timestamps in a continuous space. Details of heterogeneous graph attention can be found in Fig. 4.3.

4.6 Methods

4.6.1 Graph Node Embedding

4.6.1.1 Initial Embedding

The initial feature of a code node is parameterized by a vector $h \in \mathbb{R}^d$. These embeddings are retrieved from an embedding table $\mathbf{H}_{\mathcal{V}} \in \mathbb{R}^{d \times |\mathcal{V}|}$, with $|\mathcal{V}|$ indicating the count of unique EHR codes. This table acts as a repository for long-term collaborative signals and is refined during model training to encapsulate holistic structural information. For patient nodes, we establish a separate embedding table $\mathbf{H}_{\mathcal{U}} \in \mathbb{R}^{d \times m}$, encoding patient attributes, such as gender, where m denotes the number of distinct attribute categories. This table provides

initial states for deriving temporal embeddings for codes and patients. Throughout training, these embeddings are dynamically optimized and updated via information propagation and aggregation processes.

4.6.1.2 Temporal Embedding

Time span is essential in analyzing temporal dynamics and discovering patterns within patient EHR histories. Our approach employs a time encoding function that converts timestamps into vector representations, making it possible to depict time spans through the dot products of encoded time embeddings, thereby quantifying temporal effects in a continuous space. Specifically, the mapping $\Phi : T \rightarrow \mathbb{R}^{d_T}$ translates the time domain $T \in \mathbb{R}^+$ into a vector space of dimension d_T . In the context of sequential data modeling, it's often the relative, not absolute, time span that unveils crucial temporal dynamics. We adopt the temporal kernel function from [XRK20] to capture time span patterns. This kernel, $\mathcal{K} : T \times T \rightarrow \mathbb{R}$, is expressed as $\mathcal{K}(t_1, t_2) = \Phi(t_1) \cdot \Phi(t_2) = \psi(t_1 - t_2)$, where the dot product represents temporal correlations between two timestamps. To achieve a continuous and shift-invariant $\mathcal{K}$, as per Bochner's Theorem [Loo13], the temporal embedding $\Phi(t)$ is explicitly parameterized as

$$\sqrt{\frac{1}{d_T}} [\cos(\omega_1 t), \sin(\omega_1 t), \dots, \cos(\omega_{d_T} t), \sin(\omega_{d_T} t)], \quad (4.1)$$

with $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{d_T}]$ being learnable parameters. This methodology not only enhances our understanding of known temporal patterns but also ensures that temporal effects associated with unseen timestamps can be accurately inferred, making our model robust and capable of generalizing across any time span.

4.6.2 Time-aware Information Propagation

Our model directly processes temporal subgraphs, facilitating bidirectional information flow between patient and code nodes. At each layer, the model integrates node and time em-

beddings as input. For a given patient node u_i and query timestamp t_i , we construct a time-aware node representation for the input of the l -th layer as follows:

$$\tilde{h}_{u_i}^{(l-1)}(t_i) = h_{u_i}^{(l-1)}(t_i) + \Phi(t_i), \quad (4.2)$$

where $l = 1, 2, \dots, L$. For $l = 1$, $h_{u_i}^{(l-1)}(t_i)$ denotes the initial node embedding; for $l > 1$, it signifies the output embedding from the previous layer. In each layer, we propagate time-aware node representations of first-order neighborhood code nodes $\mathcal{N}_{u_i}^1(t_i)$ to node u_i . Along each interaction edge (u_i, v_j, t_k) , where $\max(t_k) = t_i$, the propagated information can be represented as:

$$\tilde{h}_{v_j}^{(l-1)}(t_k) = h_{v_j}^{(l-1)}(t_k) + \Phi(t_k). \quad (4.3)$$

By layering multiple levels, we capture intricate interactions among nodes connected at higher orders. This iterative propagation of information allows each patient node to receive temporal contextual information from its L -hop neighborhood code nodes, effectively modeling the temporal effect of related patient sequences on the target patient. Unlike traditional approaches that solely analyze historical EHR code impacts on patient status, our method also aims to understand the influence of patient occurrences on code attributes, extending the described information propagation mechanism to code nodes.

4.6.3 Heterogeneous Graph Attention

We propose to simultaneously model sequential patterns and temporal effects using a heterogeneous graph attention mechanism. Specifically, we apply self-attention to incorporate both structural and temporal information. For a given patient node's time-aware representation $\tilde{h}_{u_i}^{l-1}(t_i)$, information from its first-order neighborhood nodes is captured as:

$$\mathbf{Z}_{u_i}^l(t_i) = \left[\tilde{h}_{v_1}^{(l-1)}(t_1), \tilde{h}_{v_2}^{(l-1)}(t_2), \dots, \tilde{h}_{v_N}^{(l-1)}(t_N) \right], \quad (4.4)$$

The temporal embedding integrates seamlessly with the Transformer's self-attention mechanism, allowing us to convert target node embeddings into Query vectors and source

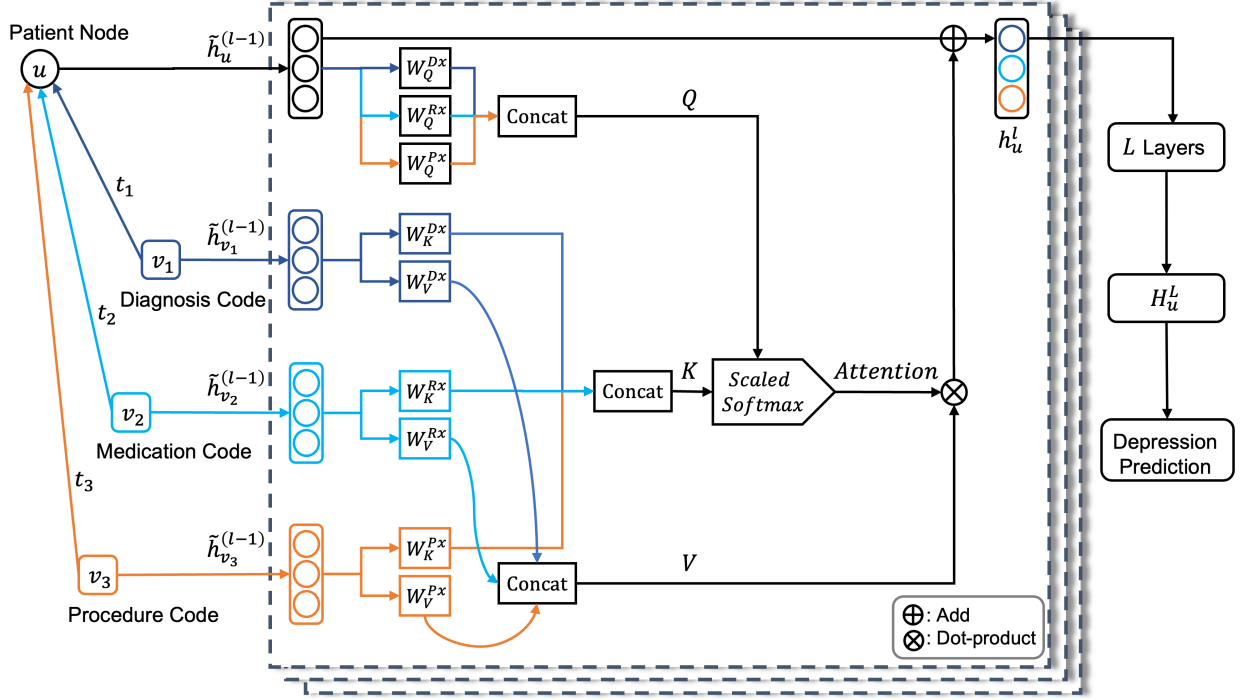


Figure 4.3: Illustration of heterogeneous graph attention mechanism.

nodes into Key vectors. Calculating their dot product yields the attention score, capturing temporal correlations between timestamps as defined by the temporal kernel function. Unlike the standard Transformer, which uses a single projection set for all inputs, our model adapts to graph heterogeneity through an edge type-dependent attention mechanism. This means employing distinct edge relations to parameterize the projection weights for each edge type, enabling nodes and edges of various types to preserve their unique representation spaces while still facilitating interaction and message exchange.

To enhance parameter sharing and preserve the distinct features of various edge relations, we derive 'query', 'key', and 'value' vectors through edge relation-specific linear projections:

$$\mathbf{q}_{u_i}^l(t_i) = \tilde{h}_{u_i}^{l-1}(t_i) \mathbf{W}_Q^{\varphi(e)},$$

$$\mathbf{K}_{u_i}^l(t_i) = \mathbf{Z}_{u_i}^l(t_i) \mathbf{W}_K^{\varphi(e)},$$

$$\mathbf{V}_{u_i}^l(t_i) = \mathbf{Z}_{u_i}^l(t_i) \mathbf{W}_V^{\varphi(e)},$$

where $\mathbf{W}_Q^{\varphi(e)}$, $\mathbf{W}_K^{\varphi(e)}$, $\mathbf{W}_V^{\varphi(e)} \in \mathbb{R}^{d \times d}$ are transformation matrices designed to represent interactions between node features and temporal embeddings. Importantly, these matrices are categorized by edge type, ensuring distinct linear projections for each interaction type to accurately model distribution variances. Subsequently, dot-product attention is utilized to calculate the weighted sum of all entities in $\mathbf{V}$:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{V} \quad (4.5)$$

where the interactions of "query-key" pairs serve as the attention weights.

To further capture the non-linear interactions between neighborhood representations and the target node features, we aggregate them by summing and then pass them to an aggregation function defined as follows:

$$h_{u_i}^l(t_i) = \text{ReLU} \left(\left[\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \tilde{h}_{u_i}^{(l-1)}(t_i) \right] \mathbf{W}^l \right) \quad (4.6)$$

where $\mathbf{W}^l \in \mathbb{R}^{d \times d}$ and $h_{u_i}^l(t_i)$ is the final output embedding of the target node u_i at time t_i in layer l . This approach is equally applicable if the query node is a code node; we simply adjust the patient query to code query information, modifying the neighbor information sequence in Eq.4.4 as needed. We update node embeddings of patients and codes nodes simultaneously at each model layer.

4.6.4 Subgraph Sampling

The essence of our method lies in explicitly capturing the complex sequence interactions among patients within temporal graphs through iterative message propagation and aggregation. As patient sequences grow, the underlying dependency graph expands significantly, posing computational challenges for conventional full-graph training approaches, especially with large-scale real-world EHR datasets. While various sampling methods have been proposed to facilitate training graph neural networks on node subsets [HYL17, ZHW19, CZS17], applying these methods directly to heterogeneous graphs often results in highly imbalanced

subgraphs. This imbalance stems from variations in degree distribution and the sheer volume of nodes across different node types. To mitigate these challenges, we introduce a neighborhood sampling strategy, enabling stochastic training on temporal heterogeneous graphs. Our method begins by choosing u_i as the root node and sampling its direct neighborhood code nodes $\mathcal{N}_{u_i}^1(t_k)$ in $\mathcal{G}^{t_k}$. To prevent potential imbalances, we implement an edge-type dependent sampling threshold to minimize information loss and sample variance across code categories. For each sampled code node $v_j \in \mathcal{N}_{u_i}^1(t_k)$, we further incorporate its first-order patient neighbors $\mathcal{N}_{v_j}^1(t_k)$, which also constitute u_i 's second-order neighbors $\mathcal{N}_{u_i}^2(t_k)$, into the subgraph. This process can be repeated up to the l -order neighborhood, where l is the predefined layer limit to control subgraph size, resulting in a tree-shaped subgraph stratified by node type at each layer.

4.6.5 Model Training

After integrating multiple layers of heterogeneous graph attention, our model is capable of generating final embeddings for the targeted patient nodes. Subsequently, a multilayer perceptron (MLP) neural network coupled with a sigmoid activation function is employed to assess the likelihood of patients being diagnosed with depression within specified time frames. During the training phase, the model's parameters were fine-tuned through end-to-end optimization, utilizing cross-entropy loss. The Adam optimizer [KB14] was chosen for its effectiveness in model optimization. To enhance generalization to unseen data, the model was trained in a denoising configuration, which involved random dropout of edge messages at a predetermined probability. Specifically, in the message-passing phase, the dropout of outgoing messages across multi-relational edges was implemented by setting the corresponding attention weights to zero. This approach augments the model's robustness to the presence or absence of particular edges. Additionally, standard dropout was applied to the MLP's hidden layers to further stabilize training.

4.7 Experiments

4.7.1 Experimental Setup

We implemented a 5-fold cross-validation approach to evaluate the performance of all models. We partitioned the patient node set into training and testing subsets, with a testing split of 20% per fold, ensuring stratification by labels to mitigate bias. To avoid information leakage, we constructed training graphs by masking connections to patient nodes designated for testing. Given the imbalance in our dataset, we adopted three metrics to assess predictive accuracy: Area Under the Precision-Recall Curve (AUPRC), Area Under the Receiver Operating Characteristic (AUROC), and Precision@k. The latter measures the fraction of true positives among the top k predictions ranked by likelihood according to model outputs. We report the performance of models on the test set, presenting both the average and standard deviation across all folds.

4.7.2 Baseline Methods

We compare our model’s performance against a variety of benchmark approaches, including those based on graph and transformer architectures. For graph models, we consider baselines such as GraphSAGE [HYL17], which averages neighboring node information followed by a linear projection; Relational Graph Convolutional Networks (RGCN) [SKB18], designed to manage relational diversity in heterogeneous graphs through distinct weights for each relationship type; and Graph Attention Network (GAT) [VCC17], employing an additive attention mechanism over neighbors. To ensure a fair comparison, we integrate our proposed subgraph sampling strategy and identical temporal embeddings into these graph baselines. This adaptation allows these baselines to operate within a temporal framework, optimizing their performance by utilizing temporal information. We standardized the hidden dimension across our proposed model and all graph baselines to 256. Additionally, the architecture for each graph model is consistently configured with 8 layers.

For transformer-based baseline, we utilize the bidirectional deep learning model BRLTM [MSO21], capable of processing heterogeneous EHR in a temporal framework for chronic disease prediction. BRLTM enhances temporal analysis by incorporating learnable token embeddings to represent patient ages across different visits. The BRLTM model is implemented according to its originally defined hyperparameters. Furthermore, we conduct ablation studies to dissect the impact of our method’s core components: subgraph sampling (S), temporal embedding (T), and heterogeneous representation (H). We systematically evaluate the effects of these components by creating model variants that exclude each component in turn (e.g., the model variant without subgraph sampling is denoted as Ours-S in Table 4.1). Specifically, omitting the subgraph sampling module treats each patient EHR sequence as disjoint graphs. Without temporal embedding, our model relies solely on static node features. Lastly, removing heterogeneous representation, we apply a uniform set of linear projections for self-attention across all edge types.

4.8 Results

4.8.1 Performance Comparison

Table 4.1 presents the performance of depression prediction on clean datasets for baseline methods alongside our proposed model and its variants. Our model consistently surpasses both graph-based and transformer-based baselines across all evaluation metrics and time windows. Among graph-based models, GraphSAGE employs mean pooling and can be seen as a basic counterpart to our approach by uniformly weighting temporal neighborhoods. This approach performs poorly on the defined prediction tasks, with AUPRC scores not exceeding 0.61. The RGCN improves on this by more adeptly managing relational diversity but still does not match our model performance, particularly over longer prediction windows. The GAT, employing an additive attention mechanism using a parameter vector and the LeakyReLU function, better captures interactions between nodes but falls approxi-

Table 4.1: Depression prediction performance on clean datasets.

Prediction Window	Three Months			Six months			One Year		
Models	AUPRC	AUROC	Precision@1k	AUPRC	AUROC	Precision@1k	AUPRC	AUROC	Precision@1k
GraphSAGE+S+T	0.61 ± 0.03	0.84 ± 0.03	0.83 ± 0.02	0.57 ± 0.03	0.79 ± 0.03	0.81 ± 0.03	0.52 ± 0.02	0.78 ± 0.03	0.68 ± 0.03
RGCN+S+T	0.62 ± 0.02	0.84 ± 0.01	0.85 ± 0.02	0.59 ± 0.03	0.81 ± 0.02	0.82 ± 0.02	0.55 ± 0.03	0.80 ± 0.02	0.71 ± 0.03
GAT+S+T	0.67 ± 0.02	0.87 ± 0.01	0.88 ± 0.01	0.63 ± 0.02	0.86 ± 0.01	0.85 ± 0.02	0.60 ± 0.03	0.84 ± 0.01	0.79 ± 0.02
BRLTM	0.69 ± 0.02	0.89 ± 0.01	0.88 ± 0.02	0.67 ± 0.02	0.88 ± 0.01	0.87 ± 0.02	0.65 ± 0.02	0.87 ± 0.01	0.82 ± 0.02
Ours	0.72 ± 0.02	0.91 ± 0.01	0.92 ± 0.01	0.70 ± 0.01	0.89 ± 0.01	0.90 ± 0.01	0.67 ± 0.02	0.88 ± 0.01	0.87 ± 0.02
Ours-S	0.69 ± 0.01	0.90 ± 0.01	0.89 ± 0.02	0.66 ± 0.01	0.88 ± 0.01	0.86 ± 0.01	0.63 ± 0.01	0.87 ± 0.01	0.80 ± 0.01
Ours-T	0.61 ± 0.02	0.85 ± 0.01	0.84 ± 0.02	0.60 ± 0.03	0.82 ± 0.01	0.80 ± 0.02	0.54 ± 0.03	0.80 ± 0.02	0.69 ± 0.02
Ours-H	0.71 ± 0.02	0.90 ± 0.01	0.90 ± 0.01	0.68 ± 0.03	0.88 ± 0.01	0.86 ± 0.03	0.65 ± 0.03	0.88 ± 0.02	0.83 ± 0.03

Table 4.2: Depression prediction performance on noisy datasets.

Prediction Window	Three Months			Six Months			One Year		
Models	AUPRC	AUROC	Precision@2k	AUPRC	AUROC	Precision@2k	AUPRC	AUROC	Precision@2k
GraphSAGE+S+T	0.68 ± 0.03	0.85 ± 0.03	0.82 ± 0.02	0.67 ± 0.03	0.85 ± 0.03	0.82 ± 0.03	0.60 ± 0.03	0.83 ± 0.03	0.78 ± 0.03
RGCN+S+T	0.71 ± 0.03	0.87 ± 0.03	0.83 ± 0.02	0.68 ± 0.03	0.86 ± 0.02	0.82 ± 0.03	0.63 ± 0.03	0.81 ± 0.02	0.80 ± 0.02
GAT+S+T	0.77 ± 0.01	0.91 ± 0.01	0.88 ± 0.01	0.73 ± 0.02	0.89 ± 0.01	0.87 ± 0.01	0.71 ± 0.02	0.88 ± 0.01	0.84 ± 0.02
BRLTM	0.77 ± 0.02	0.93 ± 0.01	0.89 ± 0.01	0.74 ± 0.02	0.90 ± 0.01	0.87 ± 0.01	0.72 ± 0.03	0.88 ± 0.01	0.83 ± 0.02
Ours	0.81 ± 0.01	0.92 ± 0.01	0.91 ± 0.01	0.80 ± 0.01	0.92 ± 0.01	0.90 ± 0.01	0.77 ± 0.02	0.90 ± 0.01	0.86 ± 0.02

mately 7 percentage points behind our model in AUPRC at the one-year window. Our use of dot-product for capturing pairwise interactions between hidden features and time embeddings, theoretically grounded in the concept of a temporal kernel and its inner-product decomposition, proves to be more effective. Although the transformer-based model BRLTM outperforms all graph-based baselines, it does not match the performance of our proposed model, particularly in terms of AUPRC and Precision@1000 metrics.

Table 4.1 also details the results of our ablation study, shedding light on the impact of various model components. Omitting the subgraph sampling module (Ours-S) consistently led to a 3-4% drop in AUPRC across three time windows, emphasizing its vital role in managing complex data relationships and leveraging high-order contextual information across different patient histories. Eliminating the temporal embedding component (Ours-T) led to more significant performance declines, with substantial decreases observed across all metrics and time windows. For instance, the one-year prediction AUPRC dropped from 0.67 to 0.54, highlighting the critical function of temporal embedding in leveraging time-dependent changes in patient data for long-term prediction. Removing the heterogeneous representation (Ours-H), which simplifies the model by treating all relationships uniformly, had a lesser but still noticeable impact. For example, the one-year prediction precision@1000 decreased from 0.87 to 0.83. These findings underscore the intricate interplay between temporal dynamics, heterogeneous interactions, and subgraph structures in EHR data, emphasizing the importance of holistic modeling approaches for accurate depression predictions.

Table 4.2 presents the comparative prediction performance on noisy datasets, further validating the model generalization capability in real-world settings involving heterogeneous labels. Our model outperforms all baselines in terms of AUPRC and Precision@2000 across three time windows. Notably, it exceeds the transformer-based BRLTM by over 5% in AUPRC at both six-month and one-year windows. When enhanced with our proposed subgraph sampling and temporal embedding techniques, GAT achieves performance levels comparable to BRLTM. Consistent with trends observed in experiments on clean datasets,

RGCN outperforms GraphSAGE due to its more effective modeling of heterogeneous graphs.

4.8.2 Graph Model Explainability

To demystify how our graph model makes predictions, we leverage the inherent graph attention mechanism to extract and analyze edge attention values. Specifically, we compute attention weights for each target patient node using softmax-normalized scores from the final layer of our trained model, focusing on the code-to-patient edges that update patient embeddings for final prediction. These edge attention weights indicate their importance to the model’s output. We analyze how attention weights vary with respect to time by plotting average edge attention weights against timespans. These timespans are calculated by subtracting the edge time (in quarters) from the latest code time of the target patient node, where a smaller timespan indicates more recent code. Figure 4.4 illustrates this analysis for the true positive patient cohort in the test set, showing that our model prioritizes recent records across all three prediction windows, particularly within the last two quarters. This observation is consistent with the patterns seen in the prediction results of Tables 4.1 and 4.2, where each model’s performance generally decreases as the prediction window extends further from the diagnosis time point. This trend agrees with our expectation that records closer to the diagnosis time are more informative and yield more accurate predictions.

4.9 Discussion

Our proposed graph model consistently outperforms the transformer baseline across various prediction windows on both clean and noisy datasets. Conceptually, the original GNNs can be seen as a variant of the transformer model. The primary distinction is that GNNs replace the transformer’s flexible attention mechanism with a fixed adjacency matrix. Conversely, the transformer model can be interpreted as a method for graph embedding, treating all nodes as interconnected and dynamically adjusting connection strengths during training.

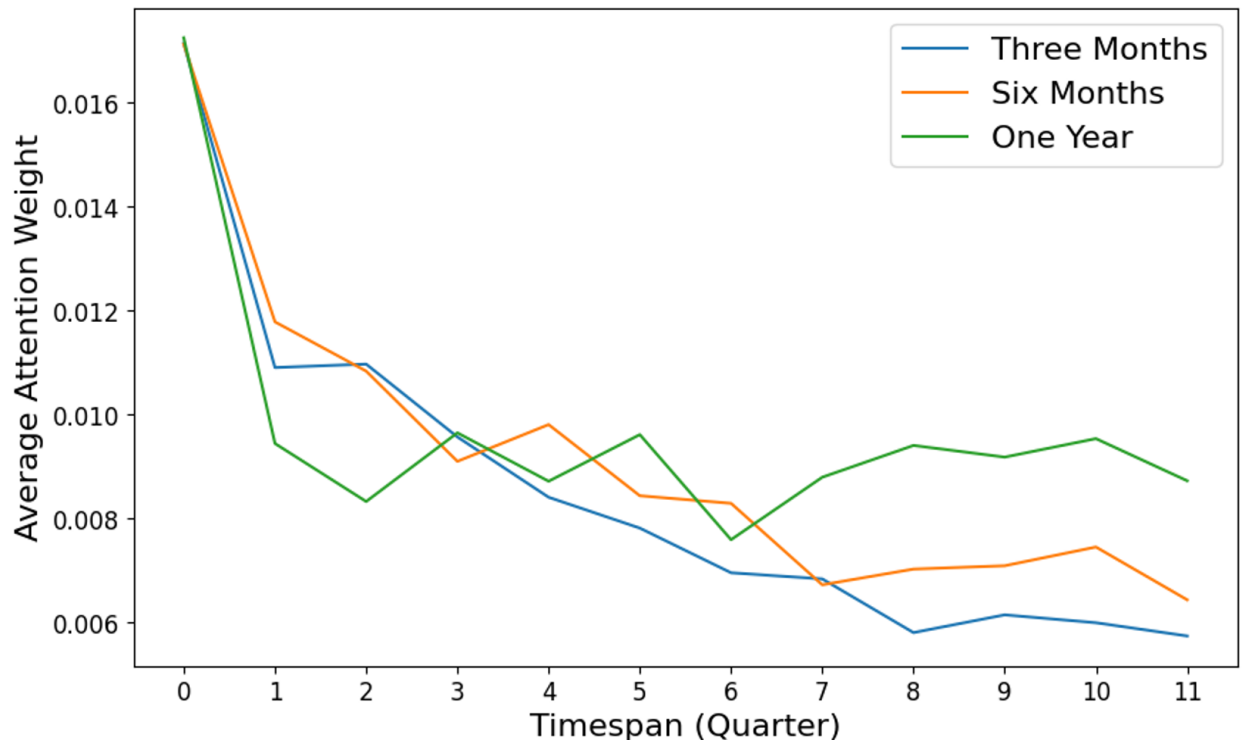


Figure 4.4: Temporal analysis of edge attention weight distribution. A smaller timespan indicates a more recent code record.

Our model synergizes the strengths of both graph and transformer architectures by incorporating a dot-product attention mechanism into the traditional graph network, while our temporal embeddings serve as a substitute for the positional encodings typically used in transformers. Moreover, our EHR graph construction introduces patient nodes and provides a more lightweight graph structure compared to the fully connected networks used in transformer models. Our heterogeneous graph attention layer substitutes the transformer’s pairwise connections with a ”gather and dispatch” mechanism, leading to more efficient processing that better handles heterogeneous information aggregation, particularly when modeling long patient sequences.

Our model’s capability to quantify temporal patterns across continuous spaces enables it to generalize and infer patient embeddings at any timestamp. This capability suggests

promising directions for future research, including the integration of our methods into a graph pretraining framework aimed at predicting next-edge (next EHR code) interactions within dynamic graphs. Leveraging graph-based pretraining techniques on more diverse and extensive EHR datasets could transform our model into a general framework applicable to a broad range of disease modeling tasks. Additionally, future research could enhance the model by integrating more data modalities, such as lab test results, to enrich the breadth and accuracy of the disease predictions.

4.10 Conclusion

In this study, we introduced a novel temporal graph learning approach that utilizes heterogeneous EHRs to improve depression prediction. Our approach integrates heterogeneous graph convolution layers with a temporal attention mechanism, allowing for the effective aggregation of temporal-topological features and high-order contextual information. This integration facilitates a nuanced understanding of patient health trajectories, enhancing both the accuracy and interpretability of predictions. Our model was rigorously evaluated on large-scale EHR datasets, where it consistently outperformed both transformer and graph-based baselines across a variety of predictive metrics and time windows. The construction of our datasets, which carefully considers the heterogeneity of depression diagnosis, coupled with our model’s robust performance on both clean and noisy datasets, underscores its suitability for real-world clinical applications. Furthermore, our approach includes a model explainability component that elucidates the temporal patterns learned by the model. In the future, we aim to expand our graph learning techniques into a comprehensive pretraining framework to increase the model’s applicability across diverse datasets and to incorporate additional data modalities. This research contributes to the advancement of clinical decision support systems for depression prediction, offering a scalable approach that facilitates timely and personalized patient care.

CHAPTER 5

Hierarchical Graph Pathomic Network for Progression Free Survival Prediction

5.1 Introduction

Prostate cancer (PCa) is the most common and second deadliest non-skin cancer in American men [SMJ19]. Prostate cancer progression following surgery is measured through changes in histological and biochemical measures. The ability to predict progression-free survival (PFS) could assist physicians in selecting treatment options for patients, with patients at a higher risk for progression receiving more aggressive treatment. The Gleason grading system is the current best method for PCa diagnosis and is a critical component in clinical survival assessment and treatment planning [EZS16]. Gleason grades are assigned by pathologists based on visual interpretation of tumor architecture. This process suffers from inter-observer variability and is constrained by its categorization system, which cannot fully capture the disease’s continuous feature spectrum. To address this issue, previous machine learning methods on survival prediction focus mainly on extracting features from manually selected regions of interest (ROIs) and use pre-defined aggregation methods to generate slide-level embeddings [CLL20, LJE19, LRW18]. However, there are two major limitations with current approaches. First, current methods ignore spatial connectivity of patches and thus fail to learn global topological representations of WSIs and may not account for tumor heterogeneity. Second, ROI-based methods often require annotations from human experts, which are expensive and can be infeasible to collect.

Deep learning methods have opened a new paradigm for automated whole slide image (WSI) analysis. However, due to image sizes, data availability, and hardware limitations, patch sampling of WSIs is necessary. To address the above challenges, graph convolutional network (GCN)-based approaches model WSIs as graph structured data, which combine global context and local connectivity with parameter efficiency. Many recent studies can be categorized into patch-based and cell-based approaches. Patch-based methods [ZYF20, LYZ18, AKT20, DLL20] focus on learning global topological representations of WSIs. For example, Li et al. [LYZ18] proposed a graph model for survival prediction by rendering the optimal graph representations of WSIs. Graph nodes were selected patches and node features were initialized by a VGG-16 network. Graph edges were constructed by thresholding the Euclidean distances between node features. Cell-based methods [ZGA19, WCL20, CLW20] aim to construct feature maps of patches by exploring the topological structure of cells and the micro-environment. As opposed to patch-based methods, cell-based methods emphasize the biological significance of WSIs. For example, Chen et al. [CLW20] proposed a multimodal fusion model for cancer survival prediction. They applied GCNs as a complementary method to convolutional neural networks (CNNs) for feature extraction. They used segmented nuclei as nodes and initialized the node attributes with handcrafted features. The edges between two nodes were thresholded by a fixed spatial distance while the maximum degree of each node was set to k corresponding to its k -nearest neighbors.

Existing GCN-based methods for WSI analysis are limited in two aspects. First, patch-based methods rely on pretrained deep learning models or handcrafted features, which ignore the informative signals from cell graphs. Second, cell-based methods are inefficient in modeling WSIs with large numbers of cells, and thus typically require laborious patch-level labeling. We propose to unify patch-based and cell-based methods with a hierarchical GCN framework, balancing efficiency and granularity. We hypothesize that analyzing WSIs through multi-scale graph representations, from fine to coarse grained, will outperform current approaches for PFS prediction using WSIs. The main contributions of our paper are

summarized as follows:

- To the best of our knowledge, we propose the first hierarchical GCN framework, termed graph pathomic network, for cancer survival prediction using WSIs, which models nuclei-level and patch-level graph structures in an integrative context.
- We propose an efficient self-supervised learning method to pretrain our graph pathomic network, yielding improved performance over trained-from-scratch counterparts.
- We develop a deep learning system for PCa PFS prediction that does not rely on manually selected ROIs and patch-level labels.

5.2 Method

5.2.1 Tumor Region Detection

Unlike previous studies, which rely on manually selected ROIs [CLL20, LJE19, LRW18], we utilize a patch-level CNN trained on weakly-labeled slides to sample cancerous patches. Specifically, we formulate the problem of training a tumor detection model with slide-level labels as a multiple instance learning (MIL) task [LLS21]. According to the basic assumption of the MIL, top patches within a malignant slide will have probabilities closer to 1, while patches from a benign slide should have probabilities closer to 0 [LLS21]. The patches with the highest predicted probabilities in each slide inherit the slide-level label. The patch-level model is further optimized with these pseudo patch-level labels. The process is iterated until convergence. The model is then finetuned on a small dataset with coarse contour annotations.

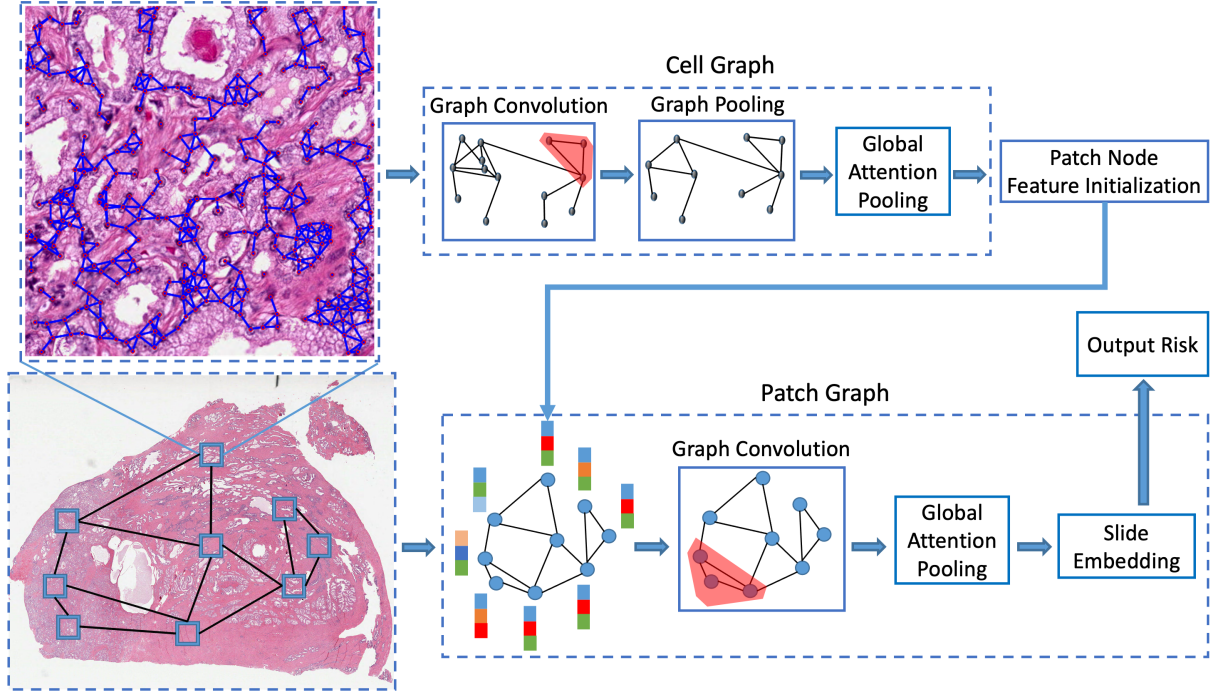


Figure 5.1: Model architecture of the hierarchical graph pathomic network. Given a WSI, we sample patches using the tumor detection model to construct a patch graph, each node of which is represented by its corresponding cell graph. Our model learns hierarchical graph representations of WSIs from nuclei-level to patch-level.

5.2.2 Graph Pathomic Network

Nuclei segmentation & Cell feature extraction. Accurate nuclei instance segmentation is important in defining reliable cell features that would be indicative of cancer progression. We train a Mask R-CNN [HGD17] using the dataset from [KVA19] to segment nuclei. After obtaining nuclei masks, we define an eight-pixel width of the ring-like neighborhood region around each nucleus as its cytoplasm area. We then extract nuclear morphometry features and imaging features (including intensity, gradient and Haralick features) for nuclear and cytoplasm regions respectively, resulting in 108-dimensional feature descriptors for each cell.

Hierarchical graph representation of WSIs.

- **Cell graph.** To learn complex relationships between cells for patch representation, we construct cell graphs by defining each cell as a node and an edge as the potential interaction between two cells. The intuition is that cells that are closer to each other are more likely to interact. We use the k-nearest neighbors algorithm to construct adjacency matrices of cell graphs. Specifically, we assign an edge between two nuclei if they are within a fixed spatial distance from each other. In addition, the maximum neighborhood number of each node is set to K corresponding to its k-nearest neighbors. We use $K=5$ to model topological information of cell graph structures, and 60 pixels at 40x resolution to set the largest distance.
- **Patch graph.** For each WSI, we define a patch graph to aggregate patch-level features and extract topological properties of the WSI. Given a set of selected patches by the tumor region detection model, we define each patch as a node in a patch graph, and initialize its features with cell graph representations. Similar to cell graph construction, we use the k-nearest neighbors algorithm ($K=3$) to construct edge sets based on spatial distance between two patches.

Graph convolutional network. We use the GCN with attention, also referred as the graph attention network (GAT) [VCC17] to perform graph convolution operations to extract and propagate features along edges of the graph. Specific model details are provided in the previous Section 2.1.2.3.

Graph pooling. To coarsen the graph and generate high-level representation, self-attention graph pooling [LLK19] based on both graph features and topology are used to reduce the number of nodes with a fixed pooling ratio k . After each graph pooling layer, the top $k\%$ nodes are selected based on their attention scores, and the remaining nodes are removed from the graph.

Model architecture. Fig. 5.1 shows an overview of our proposed model. In order to model cell-level and patch-level graph structure of WSIs in an integrative context, we build a multi-

scale GCN framework using hierarchical graph representations of WSIs. Our model consists of two consecutive steps. In the first step, our model operates directly on cell graphs of all selected patches from the same slide to generate node representations for a patch graph. We use three model layers to update node features and the topological structure of cell graphs. Each layer consists of a GAT and graph pooling operations. The graph embedding $\mathbf{r}_i$ of a cell graph is obtained by applying a global soft attention-based aggregation operation [LTB15] on updated node features $\mathbf{x}_n$, which can be expressed as:

$$\mathbf{r}_i = \sum_{n=1}^{N_i} \text{softmax}(h_{\text{gate}}(\mathbf{x}_n)) \odot \mathbf{x}_n \quad (5.1)$$

where h_{gate} is a neural network that computes attention scores by mapping node feature vectors to scalars. In the second step, we apply two GAT layers to further update the patch graph, which is initialized by features from cell graphs. Finally, we apply the same global soft attention-base aggregation operation to generate slide embeddings for PFS prediction.

5.2.3 Model Training for Progression Free Survival Prediction

We utilize the negative logarithm of Cox partial likelihood proposed in the DeepSurv model [KSC16] as the loss function for model optimization. Given the event e_i is observed for patient i at T_i , the Cox loss can be computed by Equation 5.2.

$$l(\beta) = - \sum_{i:e_i=1} (h_{\theta}(\mathbf{z}) - \log \sum_{j \in R(T_i)} \exp(h_{\theta}(\mathbf{z}))) \quad (5.2)$$

where $\mathbf{z}$ is the slide embedding of a patient, $j \in R(T_i)$ represents a set of patients with events that have not occurred at T_i . $h(\cdot)_{\theta}$ is a non-linear function (*e.g.*, a multi-layer perceptron).

5.2.4 Momentum Contrast for Unsupervised Pretraining of Cell Graph

Training an accurate GCN model often requires a large collection of labeled data and expressive features. However, we have limited patient-level labels in this study. Several recent works have demonstrated that pretrained GCN models can benefit downstream applications

by providing useful features or parameter initialization [HLG19, QCD20]. We generalize the Momentum Contrastive (MoCo) model [HFW20] to graph data for pretraining the cell graph module of our proposed framework.

The main assumption for the MoCo model is that features from augmented versions of the same input should be more similar than features from different inputs. Specifically, input data can be projected into feature vectors k_i by a key encoding function $f_k(\cdot)$, which can be considered as keys in a dataset. Given the encoded representation $\mathbf{q}_i$ of a query data generated by the query encoding function $f_q(\cdot)$, there exists one matched key, which has the largest similarity value to $\mathbf{q}_i$. The similarity between a pair of feature vectors can be measured using the cosine distance: $\text{sim}(\mathbf{q}_i, \mathbf{k}_i) = \frac{\mathbf{q}_i \mathbf{k}_i}{\|\mathbf{q}_i\| \|\mathbf{k}_i\|}$. The contrastive loss is defined by Equation 5.3.

$$\mathcal{L}_q = -\log \frac{\exp(\text{sim}(\mathbf{q}_i \mathbf{k}_i)/\gamma)}{\exp(\text{sim}(\mathbf{q}_i \mathbf{k}_i)/\gamma) + \sum_{j \neq i}^N \exp(\text{sim}(\mathbf{q}_i \mathbf{k}_j)/\gamma)} \quad (5.3)$$

where γ is the temperature parameter. $\mathbf{q}_i$ and $\mathbf{k}_i$ are feature vectors from different views of the same data i . Importantly, MoCo improves conventional contrastive learning by building a dynamic dictionary with a queue and a moving-averaged encoder. This enables building a large and consistent dictionary on-the-fly that facilitates contrastive unsupervised learning. Specifically, we define two augmentation methods to generate different views of the same cell graph. (1) Node dropping: we randomly drop cell nodes with a fixed percent number (5%) from the original graph. (2) Graph structure perturbation: we randomly set hyperparameters for cell graph construction from a fixed range of values, which also reduces model sensitivity to parameters. In this study, hyperparameters include maximum neighborhood number (5-7) and the largest distance between connected nodes (55-70 pixels).

5.3 Experiment

5.3.1 Dataset and Preprocessing

Three datasets are used in this experiment. (1) A UCLA prostate biopsy dataset containing 20,229 slides from prostate needle biopsies from 830 patients [LLS21]. (2) A Cedars-Sinai dataset consisting of 30 slides from prostatectomies of 30 patients, which were annotated with coarse contour annotations [IML18]. (3) The PFS prediction is performed on the publicly available TCGA-PRAD dataset [LLH18] using 512 * 512 pixels patches at 40x magnification (0.25 um per pixel). Datasets (1) and (2) are used to train the tumor detection model. For dataset (3), clinical data and H&E stained diagnostic slides were retrieved using the Genomic Data Commons (GDC) data portal. Additional follow-up data are obtained from the TCGA Pan-Cancer Clinical Data Resource (TCGA-CDR) [LLH18]. The recommended progression-free interval (PFI) is used as the clinical endpoint [LLH18]. 399 cases with available diagnostic slides are included in this study. Five-fold cross validation stratified by events is utilized for model training and validation. Within each fold, we further split the training data into 80% for training and 20% for validation. Model performance is evaluated on testing data for each fold to avoid overfitting.

5.3.2 Methods Comparison

To evaluate the effectiveness of our hierarchical graph pathomic network, we performed experiments on different types of features, aggregation methods, and linear Cox models.

Clinical features. We utilized clinical variables including age, PSA value, Gleason grade group (GG), and pathologic T stage to fit a baseline Cox model.

Linear Cox model. We compared our model with linear Cox models, including Lasso-Cox (*i.e.*, with $\mathcal{L}_1$ penalty) and ElasticNet-Cox (*i.e.*, with $\mathcal{L}_2$ penalty). Different types of features as described in the following sections were used as covariates for linear Cox models.

Attention MIL-based aggregation. We compared the GCN-based aggregation with the attention MIL method, which demonstrated promising results in GG prediction [LLS21]. Specifically, a trainable attention module was incorporated to summarize patch-level features into slide embeddings.

Texture-based features. We extracted 90 first-order statistics and texture-based features with the Pyradiomics package [VFP17]. Texture-based features were utilized as predictors in *ElasticNet-Cox-texture* and *Lasso-Cox-texture* experiments.

Self-supervised CNN features. We adopted the MoCo model with ResNet50 as the backbone [HFW20] to learn CNN features from patches. The trained query encoder of the MoCo model was used as the feature extractor to generate a 128×1 feature vector for each patch. Self-supervised learning features were utilized as covariates in the *Lasso-Cox-deep* and *ElasticNet-Cox-deep* experiments. *Deep-Att-MIL* denotes the deep survival model that leverages self-supervised CNN features and attention MIL-based aggregation. *Deep-GCN* denotes the variant of our proposed method, which includes only the patch graph module and uses self-supervised CNN features as initial node embeddings.

Handcrafted cell graph features. To validate the effectiveness of the cell graph, we extract pre-defined cell graph features for the ablation study. We compute global features of the nuclei with the given centroids based on the partitioning of the space into Voronoi cells and on the induced graph structure. This type of feature only captures the topological structure of cell graph.

5.3.3 Model Training and Evaluation

We selected 200 patches from each slide for analysis. We used a learning rate of 0.001 and a batch size of 64 for Deep-Att-MIL model training. The learning rate and batch size for Deep-Att-MIL-clinical were set at 0.001 and 12, respectively. All GCN-based models were trained with a batch size of 16 and a learning rate of 0.0005. Adam optimizer was used

in all experiments. PFS analysis models were evaluated with the concordance index (c-index), which was computed with the lifelines package [DKJ20]. The c-index measures the effectiveness of predicted risk scores on ranking survival times, with 1 indicating the perfect concordance and 0.5 representing results from random predictions. For cases with multiple slides, average hazards were utilized. All models were implemented with PyTorch, trained and validated with one Tesla V100 GPU.

5.4 Results

Average c-index and 95% confidence interval for model experiments are shown in the Table 5.1. The baseline Cox model with clinical features obtained an average c-index of 72.54%. The pretrained graph pathomic network that combined clinical variables achieved the highest average c-index of 79.34%, which was around 7% higher than the baseline model with only clinical features. The spatial distribution of nuclei is important for prostate progression estimation. As shown in the Table 5.1, models with handcrafted graph-based features showed better performances than those with texture-based features. Models with deep learning representations demonstrated superior performance compared with handcrafted features. The GCN based aggregation method showed better results in predicting progression compared to Att-MIL aggregation. The graph pathomic network outperformed all baseline models, demonstrating the effectiveness of learning features from hierarchical graphs. Graph MoCo pretraining brought around 1% c-index improvement than trained-from-scratch counterparts. We also observed that pretrained graph pathomic network converged faster during model training.

In summary, we developed a deep learning system with novel hierarchical graph-based representations for PFS prediction of PCa patients. By building hierarchical graph representations, our proposed model shows promising results on predicting PFS. In addition, combining clinical factors and pathomic features is superior to the baseline model with clin-

Table 5.1: Concordance index of models on predicting progression free survival for prostate cancer patients following radical prostatectomy.

Models	c-index $\uparrow$
Clinical features	0.7254 ± 0.042
Lasso-Cox-texture	0.6818 ± 0.067
Lasso-Cox-graph	0.6857 ± 0.092
Lasso-Cox-deep	0.7046 ± 0.111
ElasticNet-Cox-texture	0.6671 ± 0.052
ElasticNet-Cox-graph	0.6892 ± 0.084
ElasticNet-Cox-deep	0.7254 ± 0.101
Deep-Att-MIL	0.7420 ± 0.089
Deep-GCN	0.7441 ± 0.123
Graph pathomic network	0.7539 ± 0.103
Pretrained graph pathomic network	0.7636 ± 0.091
Deep-Att-MIL + clinical	0.7586 ± 0.076
Deep-GCN + clinical	0.7743 ± 0.122
Graph pathomic network + clinical	0.7831 ± 0.105
Pretrained graph pathomic network + clinical	0.7934 ± 0.082

ical features only, suggesting the potential of leveraging quantitative pathomic features for better progression prediction. However, we only evaluated our model on the TCGA-PRAD dataset. Validation on large prospective datasets is needed to further validate pathomic features. Reinforcement learning may also be investigated in future work to enable adaptive patch selection and end-to-end training.

CHAPTER 6

Graph Deep Learning for Tumor-associated Stroma Identification in Prostate Histopathology Slides

6.1 Introduction

Prostate cancer (PCa) diagnosis and grading rely on histopathology analysis of biopsy slides [MFH18]. However, prostate biopsies are known to have sampling error as PCa is heterogenous and commonly multifocal, meaning cancer lesions can be missed during the biopsy procedure [MLM12]. If significant PCa is detected on biopsies and the patient has organ-confined cancer with no contraindications, radical prostatectomy (RP) is the standard of care [HJG20, KPP20]. Following RP, the prostate is processed and slices are mounted onto slides for analysis. Radical prostatectomy histopathology samples are essential for validating the biopsy-determined grade group [NSG10, JKK19]. Analysis of whole-mount slides, meaning slides that include slices of the entire prostate, provide more precise tumor boundary detection, identification of various tumor foci, and increased tissue for identifying morphological patterns not visible on biopsy due to a larger field of view.

Field effect refers to the spread of genetic and epigenetic alterations from a primary tumor site to surrounding normal tissues, leading to the formation of secondary tumors. Understanding field effect is essential for cancer research as it provides insights into the mechanisms underlying tumor development and progression. Tumor-associated stroma, which consists of various cell types, such as fibroblasts, smooth muscle cells, and nerve cells, is an integral component of the tumor microenvironment that plays a critical role in tumor development

and progression. Reactive stroma, a distinct phenotype of stromal cells, arises in response to signaling pathways from cancerous cells and is characterized by altered stromal cells and increased extracellular matrix components [BTJ20, BR12]. Reactive stroma is often associated with tumor-associated stroma and is thought to be a result of field effects in prostate cancer. Altered stroma can create a pro-tumorigenic environment by producing a multitude of chemokines, growth factors, and releasing reactive oxygen species [LN18, LCT19], which can lead to tumor development and aggressiveness [HWC01]. Therefore, investigating the histological characterization of tumor-associated stroma is crucial in gaining insights into the field effect and tumor progression of prostate cancer.

Manual review for tumor-associated stroma is time-consuming and lacks quantitative metrics [DSR17, EMP18]. Several automated methods have been applied to analyze the tumor-stroma relationship; however, most of them focus on identifying a tumor-stroma ratio rather than finding reactive stroma tissue or require pathologist input. Machine learning algorithms have been used to quantify the percentage of tumor to stroma in bladder cancer patients, but required dichotomizing patients based on a threshold [ZJN23]. Software has been used to segment tumor and stroma tissue in breast cancer patient samples, but the method required constant supervision by a pathologist [MBB20]. Similarly, Akoya Biosciences Inform software was used to quantify reactive stroma in PCa, but this method required substantial pathologist input to train the software [RGD22]. Fully automated deep-learning methods have been developed to identify tumor-associated stroma in breast cancer biopsies, achieving an AUC of 0.962 in predicting invasive ductal cancer [EMP18]. However, identifying tumor-associated stroma in prostate biopsies and whole-mount histopathology slides remains challenging.

Analyzing tumor-associated stroma in prostate cancer requires combining whole-mount and biopsy histopathology slides. Biopsy slides provide information on the presence of PCa, while whole-mount slides provide information on the extent and distribution of PCa, including more information on tumor-associated stroma. Combining the information from both

modalities can provide a more accurate understanding of the tumor microenvironment. In this work, we explore the field effect in prostate cancer by analyzing tumor-associated stroma in multi-modal histopathological images. Our main contributions can be summarized as follows:

- To the best of our knowledge, we present the first deep-learning approach to characterize prostate tumor-associated stroma by integrating histological image analysis from both whole-mount and biopsy slides. Our research offers a promising computational framework for in-depth exploration of the field effect and cancer progression in prostate cancer.
- We proposed a novel approach for stroma classification with spatial graphs modeling, which enable more accurate and efficient analysis of tumor microenvironment in prostate cancer pathology. Given the spatial nature of cancer field effect and tumor microenvironment, our graph-based method offers valuable insights into stroma region analysis.
- We developed a comprehensive pipeline for constructing tumor-associated stroma datasets across multiple data sources, and employed adversarial training and neighborhood consistency regularization techniques to learn robust multimodal-invariant image representations.

6.2 Method

6.2.1 Stroma Tissue Segmentation

Accurately analyzing tumor-associated stroma requires a critical pre-processing step of segmenting stromal tissue from the background, including epithelial tissue. This segmentation task is challenging due to the complex and heterogeneous appearance of the stroma. To ad-

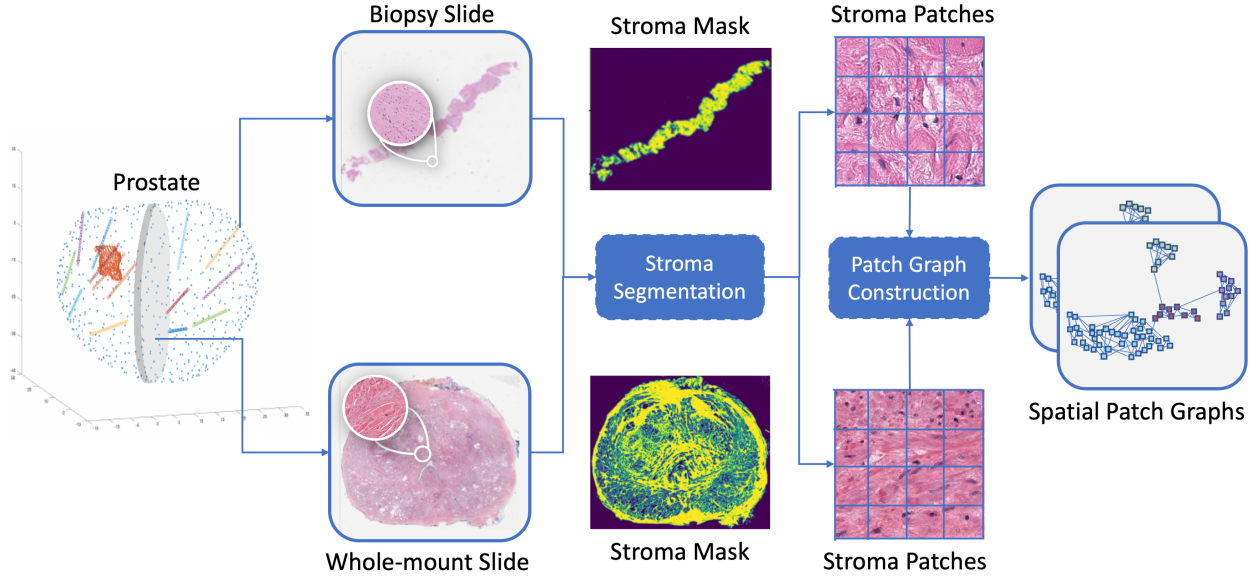


Figure 6.1: Process of stroma patch graph construction: The left prostate model illustrates the locations of biopsy and whole-mount tissues in 3D space. Biopsy slides provide a targeted view while whole-mount slides offer a broader perspective of the tumor and surrounding tissue. The stroma segmentation module generates a stroma mask to isolate the stromal tissue, which is then used to construct spatial patch graphs for the proposed deep-learning model.

dress this, we propose utilizing the PointRend model [KWH20], which can handle complex shapes and appearances and produce smooth and accurate segmentations through iterative object boundary refinement. Moreover, the model’s efficiency and ability to process large images quickly make it suitable for analyzing whole-mount slides. By leveraging the PointRend model, we can generate stromal segmentation masks for more precise downstream analysis.

6.2.2 Stroma Classification with Spatial Patch Graphs

To capture the spatial nature of field effect and analyze tumor-associated stroma, modeling spatial relationships between stroma patches is essential. The spatial relationship can reveal

valuable information about the tumor microenvironment, and neighboring stroma cells can undergo similar phenotypic changes in response to cancer. Therefore, we propose using a spatial patch graph to capture the high-order relationship among stroma tissue regions. We construct the stroma patch graph using a K-nearest neighbor (KNN) graph and neighbor sampling. The KNN graph connects each stroma patch to its K nearest neighboring patches. Given a central stroma patch, we iteratively add neighboring patches to construct the patch graph until we reach a specified layer number L to control the subgraph size. This process results in a tree-like subgraph with each layer representing a different level of spatial proximity to the central patch. The use of neighbor sampling enables efficient processing of large images and allows for stochastic training of the model.

To predict tumor-associated binary labels of stroma patches, we employ a message-passing approach that propagates patch features in the spatial graph. To achieve this, we use Graph Convolutional Networks with attention, also known as Graph Attention Networks (GATs) [VCC17]. The GAT uses an attention mechanism on node features to construct a weighting kernel that determines the importance of nodes in the message-passing process; details are provided in the previous Section 2.1.2.3. In our case, the patch graph $\mathcal{G}$ is constructed using the stroma patches as vertices, and we connect the nodes with edges based on their spatial proximity. Each vertex v_i is associated with a feature vector $\vec{h}_{vi} \in \mathbb{R}^N$, which is first extracted by Resnet-50 model [HZR16]. The final output of GAT module is the tumor-associated probability of the input patch. And the module was optimized using the cross-entropy loss L_{GAT} in an end-to-end fashion.

6.2.3 Neighbor Consistency Regularization for Noisy Labels

The labeling of tumor-associated stroma can be affected by various factors, which can result in noisy labels. One of the reasons for noisy labels is the irregular distribution of the field effect, which makes it challenging to define a clear boundary between the tumor-associated and normal stroma regions. Additionally, the presence of tumor heterogeneity and the varied

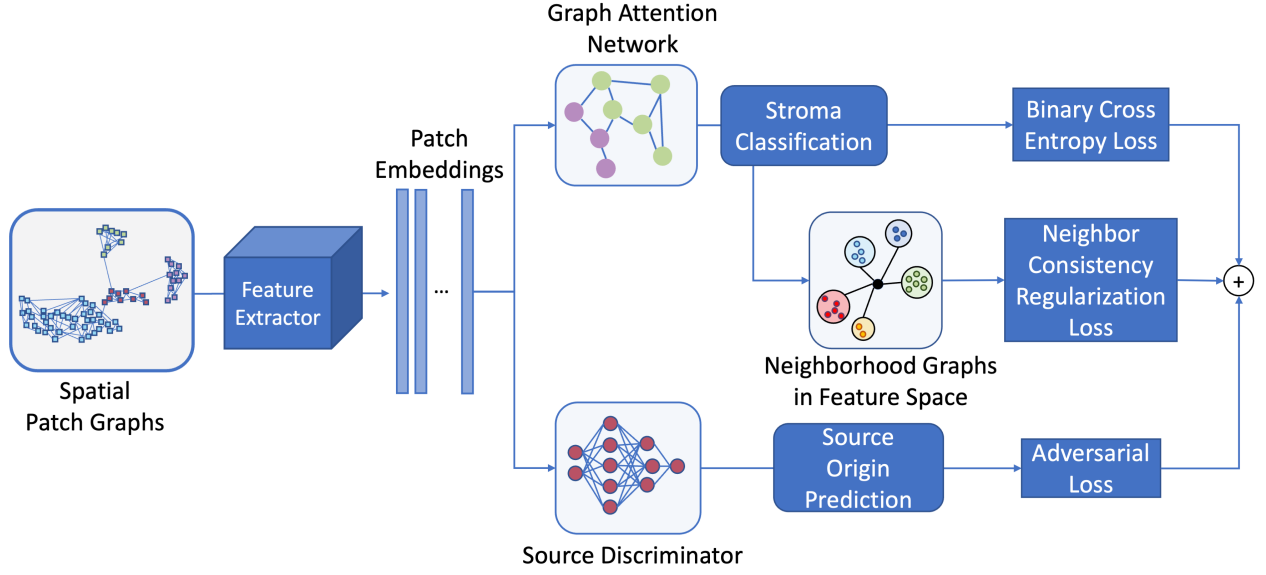


Figure 6.2: Overview of the proposed model for identifying tumor-associated stroma in multi-modal prostate histopathology slides: The input patches are represented as spatial graphs and passed through a feature extractor. The patch embeddings are fed into a graph attention network (GAT) module to capture inter-patch relationships and refine the features with neighborhood consistency regularization (NCR) for handling noisy labels. The source discriminator serves as adversarial multi-modal learning (AML) module to predict data source (biopsy/whole-mount). The stroma classifier and source discriminator are trained simultaneously with the goal of successfully classifying tumor-associated stroma using multimodal-invariant features.

distribution of tumor foci can further complicate the labeling process.

To address this issue, we propose applying Neighbor Consistency Regularization (NCR) [IVA22] to prevent the model from overfitting to incorrect labels. The assumption is that overfitting happens to a lesser degree before the final classifier, and this is supported by MOIT [OAA21], which suggests that feature representations are capable of distinguishing between noisy and clean examples during model training. Based on this assumption, NCR introduces a neighbor consistency loss to encourage similar predictions of stroma patches that

are similar in feature space. This loss penalizes the divergence of a patch prediction from a weighted combination of its neighbors' predictions in feature space, where the weights are determined by their feature similarity. Specifically, the loss function is designed as follows:

$$L_{\text{NCR}} = \frac{1}{m} \sum_{i=1}^m D_{\text{KL}} \left(\sigma(\mathbf{z}_i/T) \parallel \sum_{j \in \text{NN}_k(\mathbf{v}_i)} \frac{s_{i,j}}{\sum_k s_{i,k}} \cdot \sigma(\mathbf{z}_j/T) \right) \quad (6.1)$$

where D_{KL} is the KL-divergence loss to quantify the discrepancy between two probability distributions, T represents the temperature and $\text{NN}_k(\mathbf{v}_i)$ is the set of k nearest neighbors of v_i in the feature space.

6.2.4 Adversarial Multi-modal Learning

Biopsy and whole-mount slides provide complementary multi-modal information on the tumor microenvironment, and combining them can provide a more comprehensive understanding of tumor-associated stroma. However, using data from multiple modalities can introduce systematic shifts, which can impact the performance of a deep learning model. Specifically, whole-mount slides typically contain larger tissue sections and are processed using different protocols than biopsy slides, which can result in differences in image quality, brightness, and contrast. These technical differences can affect the pixel intensity distributions of the images, leading to systematic shifts in the features that the deep learning model learns to associate with tumor-associated stroma. For instance, a model trained on whole-mount slides only may not generalize well to biopsy slides due to systematic shifts, hindering model performance in the clinical application scenario.

To address the above issues, we propose an Adversarial Multi-modal Learning (AML) module to force the feature extractor to produce multimodal-invariant representations on multiple source images. Specifically, we incorporate a source discriminator adversarial neural network as auxiliary classifier. The module takes the stroma embedding as an input and predicts the source of the image (biopsy or whole-mount) using Multilayer Perceptron (MLP) with cross-entropy loss function L_{AML} . The overall loss function of the entire model is

computed as:

$$L_{\text{Total}} = L_{\text{GAT}} + \alpha \cdot L_{\text{NCR}} - \beta \cdot L_{\text{AML}} \quad (6.2)$$

where hyper-parameters α and β control the impact of each loss term. All modules were concurrently optimized in an end-to-end manner. The stroma classifier and source discriminator are trained simultaneously, aiming to effectively classify tumor-associated stroma while impeding accurate source prediction by the discriminator. The optimization process aims to achieve a balance between these two goals, resulting in an embedding space that encodes as much information as possible about tumor-associated stroma identification while not encoding any information on the data source. By adopting the adversarial learning strategy, our model can maintain the correlated information and shared characteristics between two modalities, which will enhance the model’s generalization and robustness.

6.3 Experiment

6.3.1 Dataset

In our study, we utilized three datasets for tumor-associated stroma analysis. (1) Dataset A comprises 513 tiles extracted from the whole mount slides of 40 patients, sourced from the archives of the Pathology Department at Cedars-Sinai Medical Center (IRB# Pro00029960). It combines two sets of tiles: 224 images from 20 patients featuring stroma, normal glands, low-grade and high-grade cancer [GIM15], along with 289 images from 20 patients with dense high-grade cancer (Gleason grades 4 and 5) and cribriform/non-cribriform glands [IML18]. Each tile measures 1200×1200 pixels and is extracted from whole slide images captured at 20x magnification (0.5 microns per pixel). The tiles were annotated at the pixel-level by expert pathologists to generate stroma tissue segmentation masks and were cross-evaluated and normalized to account for stain variability. (2) Dataset B included 97 whole mount slides with an average size of over 174,000×142,000 pixels at 40x magnification. The prostate tissue within these slides had an average tumor area proportion of 9%, with an average tumor area

of 77 square mm. An expert pathologist annotated the tumor region boundaries at the region-level, providing exhaustive annotations for all tumor foci. (3) Dataset C comprised 6134 negative biopsy slides obtained from 262 patients’ biopsy procedures, where all samples were diagnosed as negative. These slides are presumed to contain predominantly normal stroma tissues without phenotypic alterations in response to cancer.

Dataset A was utilized for training the stroma segmentation model. Extensive data augmentation techniques, such as image scaling and staining perturbation, were employed during the training process. The model achieved an average test Dice score of 95.57 ± 0.29 through 5-fold cross-validation. This model was then applied to generate stroma masks for all slides in Datasets B and C. To precisely isolate stroma tissues and avoid data bleeding from epithelial tissues, we only extracted patches where over 99.5% of the regions were identified as stroma at 40X magnification to construct the stroma classification dataset.

For positive tumor-associated stroma patches, we sampled patches near tumor glands within annotated tumor region boundaries, as we presumed that tumor regions represent zones in which the greatest amount of damage has progressed. For negative stroma patches, we calculated the tumor distance for each patch by measuring the Euclidean distance from the patch center to the nearest edge of the labeled tumor regions. Negative stroma patches were then sampled from whole mount slides with a Gleason Group smaller than 3 and a tumor distance larger than 5 millimeters. This approach aims to minimize the risk of mislabeling tumor-associated stroma as normal tissue. Setting a 5mm threshold accounts for the typically minimal inflammatory responses induced by prostate cancers, particularly in lower-grade cases. To incorporate multi-modal information, we randomly sampled negative stroma patches from all biopsy slides in Dataset C. Overall, we selected over 1.1 million stroma patches of size 256×256 pixels at 40x magnification for experiments. During model training and testing, we performed stain normalization and standard image augmentation methods.

6.3.2 Model Training and Evaluation

For constructing KNN-based patch graphs, we limited the graph size by setting $K = 4$ and layer number $L = 3$. We controlled the strength of the NCR and AML terms by setting $\alpha = 0.25$ and $\beta = 0.5$, respectively. The Adam optimizer with a learning rate of 0.0005 was used for model training. All models were implemented using PyTorch on a single Tesla V100 GPU. To evaluate the model performance, we perform 5-fold cross-validation, where all slides are stratified by source origin and divided into 5 subsets. In each cross-validation trial, one subset was taken as the test set while the remaining subsets constituted the training set. We measure the prediction performance using the area under the receiver operating characteristic (AUROC), F1 score, precision, and recall.

6.4 Results and Discussions

Table 6.1: Performance comparison with model variants. Results are averaged over 5 folds and shown in terms of mean value $\pm$ standard deviation.

Methods	AUROC	F1	Precision	Recall
Base	76.49 ± 2.19	68.29 ± 1.04	69.73 ± 1.27	66.93 ± 1.39
Base+GAT	80.02 ± 1.96	72.56 ± 0.72	73.69 ± 0.63	71.47 ± 1.21
Base+GAT+AML	85.74 ± 1.18	76.73 ± 1.10	76.47 ± 1.55	77.02 ± 1.54
Base+GAT+AML+NCR	86.53 ± 0.38	79.26 ± 0.36	78.45 ± 0.49	80.10 ± 0.32

To evaluate the effectiveness of our proposed method, we conducted an ablation study by comparing the performance of different model variants presented in Table 6.1. Specifically, the base model is the ResNet-50 feature extractor for tumor-associated stroma classification. Each model variant included a different combination of modules presented in method sections. We systematically add one or more modules to the base model to evaluate their performance contribution. The results show that the full model outperforms the base model

by a large margin with 10.04% in AUROC and 10.97% in F1 score, and each module contributes to the overall performance. Compared to the base model, the addition of the GAT module resulted in a significant improvement in all metrics, suggesting spatial information captured by the patch graph was valuable for stroma classification. The most notable performance improvement was achieved by the AML module, with a 5.72% increase in AUROC and 5.55% increase in Recall. This improvement indicates that AML helps the model better capture the multimodal-invariant features that are associated with tumor-associated stroma while reducing the false negative prediction by eliminating the influence of systematic shift cross modalities. Finally, the addition of the NCR module further increased the average model performance and improved the model robustness across 5 folds. This suggests that NCR was effective in handling noisy labels and improving model’s generalization ability.

In conclusion, our study introduced a deep learning approach to accurately characterize the tumor-associated stroma in multi-modal prostate histopathology slides. Our experimental results demonstrate the feasibility of using deep learning algorithms to identify and quantify subtle stromal alterations, offering a promising tool for discovering new diagnostic and prognostic biomarkers of prostate cancer. Through exploring field effect in prostate cancer, our work provides a computational system for further analysis of tumor development and progression. Future research can focus on validating our approach on larger and more diverse datasets and expanding the method to a patient-level prediction system, ultimately improving prostate cancer diagnosis and treatment.

CHAPTER 7

Self-supervised Learning for Digital Pathology using Vector-Quantized Image Modeling

7.1 Introduction

Computational pathology using digitized slide images has emerged as a valuable yet challenging domain in medical image analysis. Advances in deep learning have positioned computational pathology as a key enabler of precision medicine, optimizing cancer diagnosis and treatment strategies. However, the field faces significant challenges, particularly the scarcity of data and the high costs associated with obtaining detailed annotations. Self-supervised learning (SSL) has recently shown promise in effectively learning foundational data representations, enhancing the generalizability of models for downstream tasks. Drawing inspiration from recent developments in foundational vision models for natural images—specifically vector-quantized modeling and generative pretraining—we developed an SSL framework tailored for computational pathology. Our approach utilizes a comprehensive, standardized dataset encompassing over 48 organs and nearly 100 disease types. The model learns a rich perceptual codebook to encode diverse pathology phenotypes. Building on this domain-specific image tokenizer, we further develop a Generative Pretrained Transformer (GPT) to enhance generalizable pathology image composition modeling.

Unlike natural images, digital pathology slide images are characterized by their gigapixel resolution and hierarchical structure. Image features at various magnification levels are complementary, offering distinctive signals for different tasks. Existing SSL models for digital

pathology often neglect this aspect, typically pretraining with images at limited magnifications or using input-space augmentations to implicitly address the variable magnification in the dataset. Such models fail to capture magnification-specific information and do not offer effective fine-tuning and adaptation strategies for downstream tasks across magnification levels. Conversely, our model incorporates scale-aware positional encoding, enabling explicit conditioning on the magnification level of input data, which enhances performance on downstream tasks. We have also developed efficient fine-tuning strategies to adapt the pretrained model across multiple magnification levels, positioning our model as a versatile backbone for rapid adaptation in future research. Extensive testing on various downstream tasks demonstrates that our model achieves competitive results compared to state-of-the-art fully supervised methods. Furthermore, our model exhibits impressive capabilities in realistic image synthesis and superior generalization, particularly when labeled data is limited. This study advances self-supervised learning in digital pathology, both in terms of pretraining approaches and downstream evaluations, enabling a data-efficient deep learning model capable of generalizing and transferring across diverse image analysis tasks and clinical workflows.

7.2 Methods

7.2.1 Whole Slide Image Representation

The development of whole slide image (WSI) scanners has revolutionized the digitization of glass slides, enabling high-resolution imaging. Typically, a 20mm x 15mm sample is digitized at a resolution of 0.25 micrometers per pixel (MPP). Conventionally, optical microscopes use a 10X eyepiece magnification, which when combined with a 40X objective lens, achieves a 400X total magnification level. Images captured at 0.25 MPP are typically referred to as 40X, while those at 0.5 MPP are considered 20X. This high-resolution baseline makes WSIs particularly large, often reaching 100,000 x 100,000 pixels in size. In clinical practice, pathologists typically begin diagnoses by examining a slide at 4X magnification to

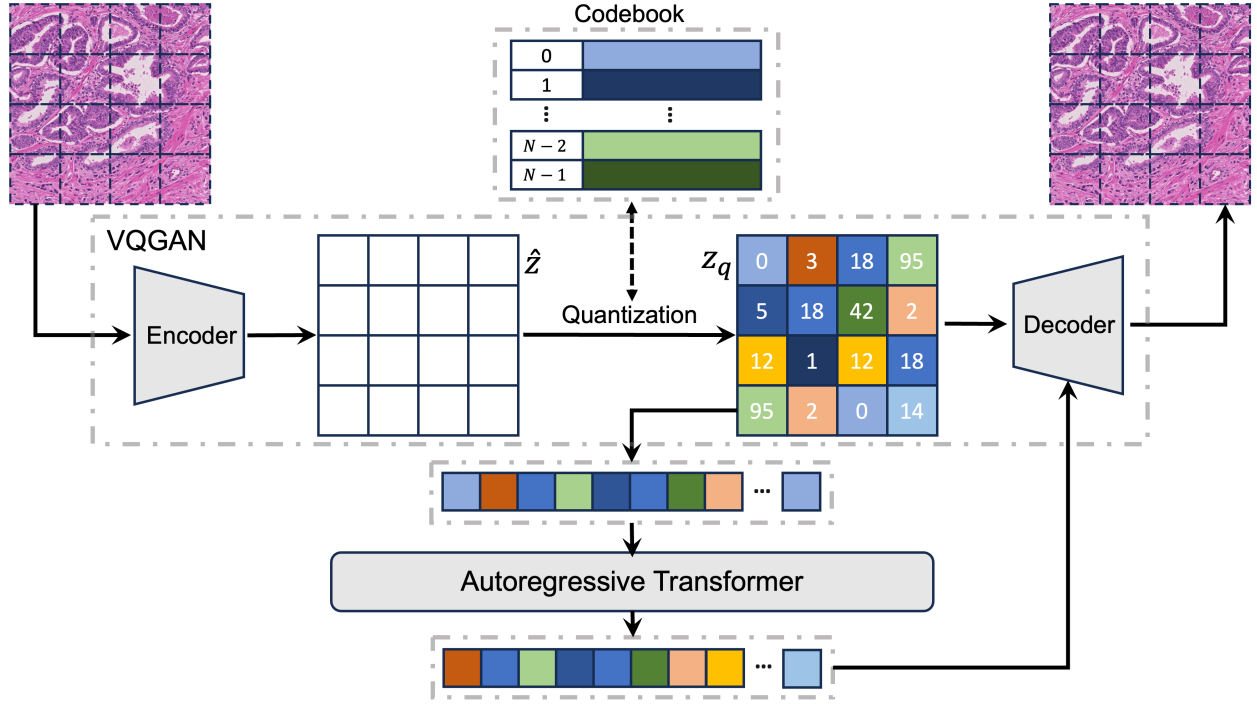


Figure 7.1: Overview of the proposed self-supervised learning model architecture. A VQGAN is utilized to learn a codebook of perceptually rich visual tokens from diverse pathology image datasets. This learned discrete codebook serves as the interface between the VQGAN and a latent Transformer. The Transformer models image composition through autoregressive prediction of the next image token.

identify any tumors or lesions. Further analysis at 10X allows for detailed observations of tissue morphology and edges, while 20X is used to examine cellular clusters and intercellular relationships more closely.

Due to their gigapixel size, WSIs present challenges in analysis and display, limited by computer memory and screen size. To accommodate this, WSIs are typically stored in a multi-resolution pyramidal format, where images at various resolutions are organized into a hierarchical structure of split tiles at different levels. Let $I \in \mathbb{R}^{H \times W \times 3}$ represent an input RGB image of a tile, with height H and width W . Our model patchifies I into a sequence S of independent patches of height and width P pixels. Specifically, in our experiments, we

split a 256×256 pixel tile into 16×16 non-overlapping patches with size of 16×16 . We first employ a visual tokenizer module to encode these patches into a sequence of 256 tokens using a learned discrete codebook. We then train a Transformer to model the image composition by predicting the next token index autoregressively. Furthermore, we incorporate scale-aware position encodings with token embeddings to capture the spatial location and resolution of each patch during Transformer training.

7.2.2 Learning Visual Tokenizer using VQGAN

We tokenize the input image into a sequence of semantic tokens using a VQGAN tokenizer, which allows our model to operate on semantic tokens rather than raw pixels. This approach enables the use of the expressive Transformer architecture for scalable pretraining. Following the methodology of [ERO21], we learn a discrete spatial codebook using a convolutional model with an encoder E and a decoder G . Specifically, we employ a UNet model used by [RBL22] to approximate an input image x with $\hat{x} = G(z_{\mathbf{q}})$, using a learned codebook $\mathcal{Z} = \{z_k\}_{k=1}^K \subset \mathbb{R}^{n_z}$, where K is the vocabulary size and n_z is the dimension of latent codes. We obtain $z_{\mathbf{q}}$ using element-wise quantization $\mathbf{q}(\cdot)$ of each spatial code $\hat{z}_{ij} \in \mathbb{R}^{n_z}$ onto its closest codebook element z_k :

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg \min_{z_k \in \mathcal{Z}} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{(H/P) \times (W/P) \times n_z} \quad (7.1)$$

The decoder reconstructs x from this quantized latent space. The model and codebook are trained end-to-end using a loss function that combines reconstruction loss and commitment loss to stabilize training:

$$\begin{aligned} \mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) = & \|x - \hat{x}\|^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 \\ & + \beta \|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2 \end{aligned} \quad (7.2)$$

where β is the weighting factor and sg denotes the stop-gradient operations. To further enhance the perceptual quality and compression rate of the codebook, an adversarial training loss was introduced with a patch-based discriminator D that discriminates between real and

reconstructed image patches. The complete training objective is:

$$\begin{aligned} \arg \min_{E, G, \mathcal{Z}} \max_D \mathbb{E}_{x \sim p(x)} [\mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) \\ + \lambda [\log D(x) + \log(1 - D(\hat{x}))]] \end{aligned} \quad (7.3)$$

To integrate context from all parts of the image, we incorporate a single attention layer at the bottleneck’s lowest resolution within the UNet. This training procedure significantly shortens the sequence length of the latent code, enabling the application of scalable Transformer models.

7.2.3 Modeling Image Composition with Scale-aware Transformer

7.2.3.1 Latent Transformer

Utilizing a learned VQGAN, we convert images into sequences of codebook indices representing their discrete latent embeddings. Specifically, the quantized encoding of an image x corresponds to a sequence $s \in \{0, \dots, |\mathcal{Z}| - 1\}^{h \times w}$ of codebook indices, which is obtained by replacing each code with its index in the codebook $\mathcal{Z}$. Then a decoder-only Transformer model is employed to model the image composition by predict the next index autoregressively. Given indices $s_{<i}$, the transformer predicts the distribution of potential next indices $p(s_i | s_{<i})$, thereby calculating the likelihood of the entire sequence as $p(s) = \prod_i p(s_i | s_{<i})$. The training objective is to minimize the negative log-likelihood of the data representations:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)} [-\log p(s)]. \quad (7.4)$$

We adopt the GPT architecture for Transformer model. Specifically, each discrete image token index is embedded into a learnable space, augmented with a scale-aware positional encoding described in the subsequent subsection 7.2.3.2. Both the token and positional embeddings match the model’s dimensional specifications. We then process these inputs through a series of Transformer blocks employing causal attention across the entire sequence.

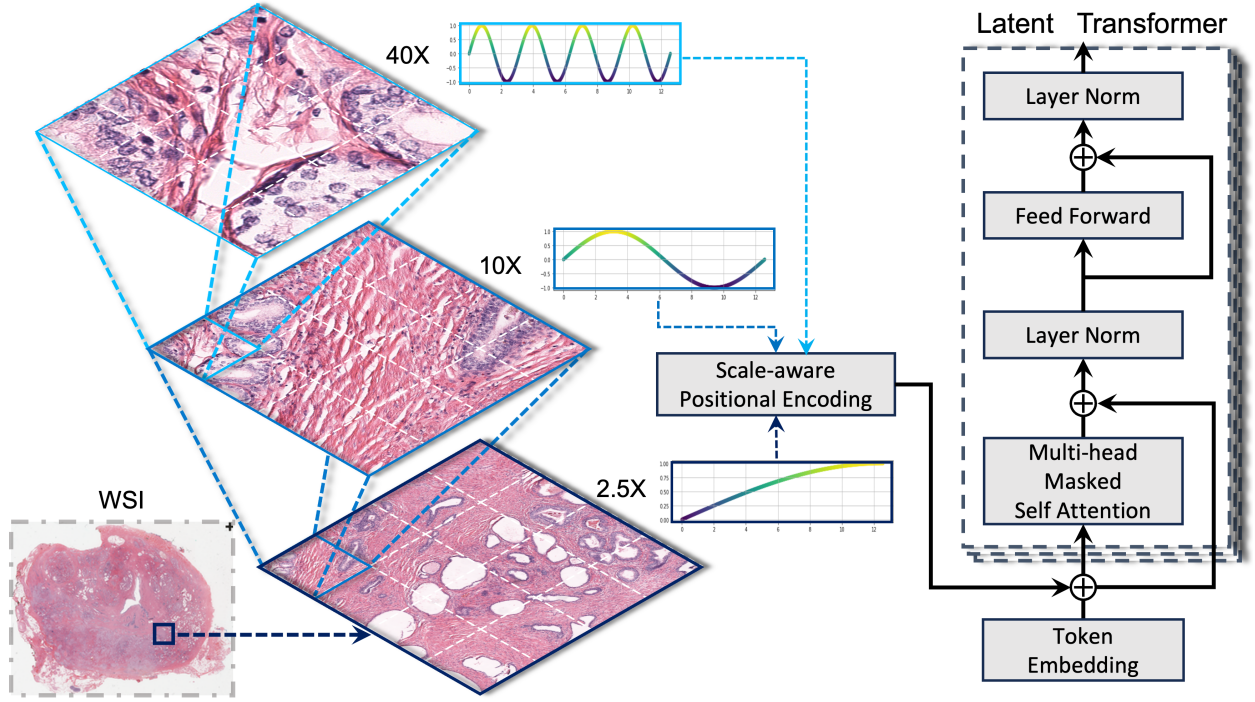


Figure 7.2: Illustration of scale-aware positional encoding for WSIs. This approach improves upon the traditional Transformer’s positional encoding by incorporating resolution information, specifically by scaling the positional encodings according to the magnification level of the input image.

7.2.3.2 Scale-aware Positional Encoding

Image tiles captured at higher magnification levels, indicated by smaller MPP, contain finer details compared to images captured at lower magnification levels with larger MPP. However, existing models often overlook the absolute scale of images when trained on pathology WSIs datasets. Even if they implicitly learn about the variable resolution present in the dataset through input-space augmentations, they fail to explicitly condition on the magnification level of unseen data. To address this limitation, we enhance the traditional Transformer’s positional encoding by incorporating resolution information. This enhancement involves scaling the positional encoding based on the magnification level of the input image. Mathe-

matically, this scaling can be expressed as:

$$\begin{aligned} v_x(pos, 2i) &= \sin \frac{M}{m} \frac{pos}{10000^{\frac{2i}{D}}} \\ v_y(pos, 2i + 1) &= \cos \frac{M}{m} \frac{pos}{10000^{\frac{2i}{D}}} \end{aligned} \tag{7.5}$$

Here, pos denotes the position of the tile along the given axis in the 2-dimensional spatial space, i represents the encoding index, M signifies the magnification level (e.g., 40X) of the image, and m stands for a reference magnification level (20X in our experiments). In essence, objects imaged at higher magnification levels are represented by more pixels, capturing finer, higher-frequency details. In contrast, at lower magnifications, these details are represented by fewer pixels, encapsulating lower-frequency information. These scale-aware positional encodings are then concatenated with and added to token embeddings during latent Transformer training.

7.2.4 Unifying Image Generation and Multi-scale Representation Learning

With a pretrained VQGAN and generative Transformer models, we can implement a unified framework for downstream tasks in image generation and representation learning.

7.2.4.1 Image Generation

Class-unconditional image generation can be accomplished by sequentially sampling from the softmax distribution of the Transformer’s output. The sampling process may begin with either a random token from the VQGAN codebook or a subset of tokens from an encoded image for image outpainting. After iterating through the generation steps, the sequence of tokens is fed into the VQGAN decoder to produce the output images.

7.2.4.2 Patch-level Representation Learning

For image understanding tasks, we input all semantic tokens produced by the VQGAN into the Transformer, generating a sequence of token features. In line with prior studies [CRC20, YLK21], we select outputs from a specific block within the Transformer as the image representation. This approach avoids the complexity of concatenating outputs from multiple layers, enabling us to use either the entire set of layers or a selected subset as the pretrained backbone for feature extraction. These features are then adapted using two finetuning strategies for different downstream tasks, as outlined in Section 7.3.2. Consistent with findings from previous research [CRC20, YLK21], we find that representations from the middle Transformer layers yield higher linear probing accuracy for downstream classification tasks. A more thorough analysis and details on the experimental setups are available in Section 7.5.2.

7.2.4.3 Slide-level Representation Learning

Inspired by recent advancements in sequence length scaling for language and image modeling [DMD23, WMX23], we utilize our pretrained VQGAN tokenizer to process WSIs as extremely long sequences of patch tokens. We enhance our latent Transformer with a dilated attention mechanism, as proposed by LongNet [DMD23], enabling efficient end-to-end processing of gigapixel images. This approach effectively manages both short-range and long-range dependencies, eliminating the need for multi-stage training.

The fundamental operation within traditional Transformers is self-attention, which maps a query and a set of keys and values to output. For inputs $Q, K, V \in \mathbb{R}^{N \times d}$, the output O is calculated as:

$$O = \text{softmax}(QK^T)V \quad (7.6)$$

However, self-attention is computationally intensive for long sequences due to its quadratic dependency on sequence length. To address this, we employ dilated attention to segment

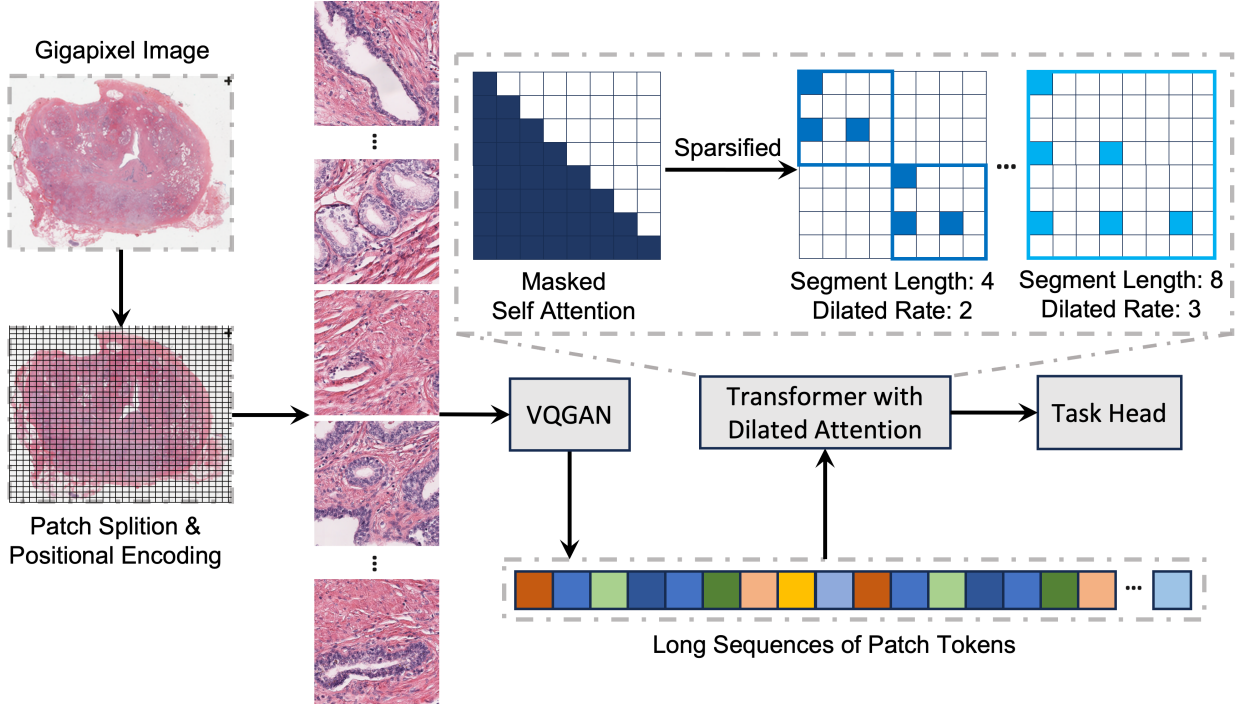


Figure 7.3: Illustration of the dilated attention mechanism for WSI modeling. This mechanism employs various attention patterns to capture both short-range and long-range dependencies. The number and hyperparameters of these patterns are adaptable based on the sequence length. This configuration enables the extension of a pretrained Transformer to model very long sequences, obtaining representations of entire gigapixel images without adding extra layers or increasing computational costs.

the inputs (Q, K, V) into smaller groups $\left\{ \left(\tilde{Q}_i, \tilde{K}_i, \tilde{V}_i \right) \right\}^{\frac{N}{w}}$ equally with a segment length w . Each segment is sparsified along the sequence dimension by selecting rows at an interval of r and processed in parallel, resulting in a concatenated output O :

$$\begin{aligned}
 \tilde{O}_i &= \text{softmax} \left(\tilde{Q}_i \tilde{K}_i^T \right) \tilde{V}_i \\
 \hat{O}_i &= \left\{ \tilde{O}_{i,j} | j \bmod r = 0; 0 | j \bmod r \neq 0 \right\} \\
 O &= \left[\hat{O}_0, \hat{O}_1, \dots, \hat{O}_{\frac{N}{w}-1} \right]
 \end{aligned} \tag{7.7}$$

Our implementation of dilated attention offers significant advantages for processing gigapixel

images: it notably reduces computational complexity from quadratic to linear and acts as a drop-in replacement for standard attention, facilitating seamless integration with our pretrained model during finetuning.

7.3 Experiment Setup

7.3.1 Model Pretraining

We pretrain a general self-supervised model using the Comprehensive Pathological Image Analysis (CPIA) dataset [YLZ23]. This dataset effectively addresses the common limitations of comprehensiveness and diversity found in existing collections, which are vital for capturing detailed pathological information across various scales. The CPIA dataset integrates 103 open-source datasets into a standardized, large-scale collection that covers over 48 organs/tissues and nearly 100 disease types. These images are organized into four distinct scales: the largest (XL) features 3840x3840 pixel patches from WSIs at a uniform MPP of 0.491, approximating a 20X objective lens. Each XL image is further segmented into 16 large (L) scale images (960x960 pixels) and 32 medium (M) scale images (384x384 pixels), with each M scale yielding a small (S) scale image (96x96 pixels).

Two versions of the CPIA dataset are available: the full CPIA, which includes all data without filtering, and the CPIA-Mini, a streamlined variant that resizes images at XL, L, and M scales to 384x384 pixels, corresponding to magnifications of 2X, 8X, and 20X. We use over 2.2 million of these patches for pretraining. The CPIA-Mini, built solely from the training sets of public datasets, guarantees no overlap with the validation and test subsets of downstream datasets, effectively preventing potential information leakage. This configuration is essential for accurately assessing the pretrained model’s effectiveness across various tasks. Furthermore, during pretraining, we apply data augmentation techniques such as random flipping, resizing, and cropping. These approaches introduce additional simulated magnifications of 5X, 10X, and 40X, thus broadening the model’s exposure to a range from

2X to 40X and improving its adaptability across different scales and contexts.

We pretrained our VQGAN using four downsampling blocks to compress input images from 256x256 to 16x16 tokens and established a codebook of 1024 unique entries, each represented by a 1024-dimensional vector. Subsequently, we trained a latent Transformer to predict token sequences of length 256 using the GPT-2 medium architecture, which includes 307 million parameters, 24 layers, a hidden size of 1024 units, and 16 self-attention heads. All pretraining was conducted on a DGX-1 server using eight NVIDIA Tesla V100 GPUs, while downstream experiments utilized two GPUs.

7.3.2 Finetune Strategies for Downstream Tasks

To evaluate our pretrained model’s performance on downstream tasks, we implement two finetuning strategies: (1) freezing the original model weights while incorporating new learnable layers, and (2) employing delta tuning using the Low-Rank Adaptation (LoRA) strategy [HSW21]. Both approaches are designed to leverage the robust foundational representations of the pretrained model while minimizing the number of trainable parameters.

7.3.2.1 Static Feature Extraction with Task-Specific Layers

In our first strategy, we maintain the integrity of the pretrained model weights to preserve essential generalized features. We adapt the model for specific tasks as follows:

- **Classification Tasks:** We adopt a linear probing protocol, commonly used in self-supervised learning evaluations, where the backbone model is kept static. Averaged patch token features are utilized as the global representation. Subsequently, a linear classification head is trained on these representations. This head is specifically tailored to accommodate the class configuration of the downstream task, effectively mapping the pretrained features to class-specific outputs.

- **Segmentation Tasks:** In our VQGAN setup, each token input to the Transformer corresponds to a 16x16 pixel subpatch. To facilitate segmentation, we build upon the methodologies established in previous works [PDB22a, RWZ23]. Specifically, we employ UperNet [XLZ18] as task layers attached to the Transformer model backbone. UperNet comprises multiple convolutional layers equipped with upsampling features. These layers are engineered to expand the feature maps generated by the pretrained model to the original input size, thereby enabling precise pixel-level predictions with spatial accuracy.

7.3.2.2 Delta Tuning with Low-Rank Adaptation

We can further enhance model adaptability by employing Delta Tuning with the Low-Rank Adaptation (LoRA) approach [HSW21]. This method involves embedding trainable low-rank matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ into the existing architecture, by updating the pretrained weight matrix $W \in \mathbb{R}^{d \times k}$ with $\Delta W = BA$, while ensuring $r \ll \min(d, k)$. Throughout the training phase, only A and B are learnable, with A initialized to a random Gaussian distribution and B set to zero. This initialization keeps ΔW zero at the start, promoting stability. The subtle alterations introduced by these matrices refine the model’s functionality for new tasks while maintaining the integrity of the pretrained weights. In our experiments, LoRA is applied at a low rank $r = 2$, targeting only the attention weights for query (W_q) and values (W_v) of each layer, thereby limiting the expansion of trainable parameters.

7.4 Experiment Results

7.4.1 Patch-level Image Classification

We assessed the patch-level representation quality of our model using two public breast cancer image datasets. The first, PCam [VLW18], was utilized for tumor presence classification at

Table 7.1: Classification of tumor presence in the PCam dataset. Top: performance comparison of CNN models with fewer than 1 million trainable parameters. Middle: evaluation of state-of-the-art models with a larger parameter budget (several million). Bottom: performance of our proposed methods utilizing the minimum number of trainable parameters.

Method	Trainable Parameters	AUC
H-Net [WGT17]	542 <i>K</i>	0.939
VF-CNN [MVK17]	556 <i>K</i>	0.953
<i>G</i> -CNN [BLV18, LBP21]	557 <i>K</i>	0.968
Steerable <i>G</i> -CNN [WHS18]	565 <i>K</i>	0.971
ResNet-34 [HZR16]	21.3 <i>M</i>	0.942
ResNet-50 [HZR16]	23.5 <i>M</i>	0.948
DenseNet-121 [HLV17]	7.8 <i>M</i>	0.921
DenseNet-169 [HLV17]	13.3 <i>M</i>	0.920
DSF-CNN [GER20]	2.2 <i>M</i>	0.975
Ours-Linear probing	2.1K	0.970
Ours-LoRA	198.7 <i>K</i>	0.979

a single magnification level. The second, BreakHis [SOP15], was used for tumor malignancy classification across multiple magnification levels.

7.4.1.1 PCam dataset

The PCam dataset comprises 327,680 image patches, each 96x96 pixels, extracted at 10X from WSIs in the Camelyon16 [BVV17] dataset, which includes 400 H&E-stained WSIs of sentinel lymph node sections. A patch is labeled as tumor-positive if its central 32x32 pixel region contains at least one tumor pixel, according to the original WSI annotations. The dataset is split into 75% training, 12.5% validation, and 12.5% testing, using a hard-negative mining approach. Following established methodologies [GER20], we report the

Table 7.2: Classification of tumor malignancy in the BreakHis dataset, comparing patient recognition rates across test image subsets at various magnifications. The methods fall into two categories: magnification-specific methods, trained and tested separately for each magnification level; and magnification-independent methods, using a single unified model trained and tested on images from all magnifications.

Category	Method	4x	10x	20x	40x	Average
Magnification Specific	Spanhol et al. [SOP15]	0.838 ± 0.041	0.821 ± 0.049	0.851 ± 0.031	0.823 ± 0.038	0.833
	Spanhol et al. [FLC16]	0.900 ± 0.067	0.884 ± 0.048	0.846 ± 0.042	0.861 ± 0.062	0.873
	Song et al. [SZC17]	0.900 ± 0.032	0.889 ± 0.050	0.869 ± 0.052	0.863 ± 0.070	0.880
	Song et al. [SCH17b]	0.900 ± 0.032	0.912 ± 0.044	0.878 ± 0.053	0.874 ± 0.072	0.891
	Han et al. [HWZ17]	0.971 ± 0.015	0.957 ± 0.028	0.965 ± 0.021	0.957 ± 0.022	0.962
	Gupta et al. [GB18]	0.947 ± 0.009	0.959 ± 0.042	0.968 ± 0.011	0.891 ± 0.001	0.941
Magnification Independent	Bayramoglu et al. [BKH16b]	0.831 ± 0.021	0.832 ± 0.035	0.846 ± 0.027	0.821 ± 0.044	0.833
	Gupta et al. [GB17]	0.872 ± 0.037	0.882 ± 0.033	0.889 ± 0.025	0.858 ± 0.038	0.875
	Ours-w/o scale-aware	0.834 ± 0.035	0.885 ± 0.006	0.929 ± 0.013	0.865 ± 0.028	0.878
	Ours-Linear probing	0.913 ± 0.004	0.907 ± 0.003	0.926 ± 0.002	0.918 ± 0.003	0.916
	Ours-LoRA	0.948 ± 0.016	0.934 ± 0.013	0.955 ± 0.015	0.933 ± 0.024	0.943

test area under the receiver operating characteristic curve (AUROC) to evaluate binary classification performance.

Table 7.1 presents comparative results between our proposed methods and several deep learning baselines across varying sizes of trainable parameters. Notably, our pretrained model with linear probing achieves test AUROC comparable to the top two performing baselines while utilizing significantly fewer trainable parameters (2.1 thousand, less than 0.37% of those in the Steerable *G*-CNN [WHS18]). Furthermore, when finetuned with the LoRA strategy, our model demonstrates superior performance, effectively utilizing 198.7 thousand trainable parameters (0.06% of the total pretrained Transformer parameters) to outperform the state-of-the-art baseline, which has 10 times more trainable parameters. These results highlight the robust generalization capability of our pretrained models and the efficiency of our finetuning strategy.

7.4.1.2 BreakHis dataset

consists of 7,909 images from 82 patients, including 24 with benign tumors and 58 with malignant tumors. Each patient’s images are available in four magnifications: 4X, 10X, 20X, and 40X. Following previous work [GB18], we randomly selected 58 patients (70%) for training and the remaining 25 (30%) for testing, across five trials of random training-testing splits to report average test performance. We also used the patient recognition rate (PRR) to evaluate classification accuracy at the patient level, determined by averaging the individual patient scores across the test cohort. Each patient score represents the ratio of correctly classified and total cancer images for that patient.

Table 7.2 presents the comparative results between our methods and two categories of baselines: magnification-specific methods, trained and tested individually for each magnification level, and magnification-independent methods, utilizing a single unified model for all magnifications. Magnification-specific baselines generally outperformed the magnification-independent baselines, suggesting the challenge of a single model learning discriminative patterns at varying magnifications. Nevertheless, our pretrained model with linear probing achieved promising results, outperforming all magnification-independent baselines and improving the average PRR by 4.1%. While our pretrained model finetuned with LoRA did not surpass the best magnification-specific baseline, it delivered competitive performance, ranking within the top three across all magnifications.

Further ablation study was conducted on the impact of our proposed scale-aware positional encoding. We trained a variant using traditional positional encoding with the same VQGAN and Transformer architecture (denoted as Ours-w/o scale-aware in Table 7.2). Comparatively, our model with linear probing exceeded the variant by 3.8% in average PRR, notably outperforming at magnifications of 4X and 40X by 7.9% and 5.3%, respectively. These findings confirm that our scale-aware positional encoding enhances the Transformer’s ability to process and apply multi-magnification patterns in diverse pathology images effec-

tively. Compared to the magnification-specific approach, which requires extensive training and adaptation, our method offers robustness and flexibility, making it better suited for real clinical settings where it can accommodate the variability in pathology images across different magnifications.

7.4.2 Patch-level Semantic Segmentation

We conducted our experiments on two histopathology image segmentation datasets focused on colorectal and prostate cancers: the GlaS dataset [SPC17] and the Prostate Gleason Grading dataset [GIM15, IML18].

7.4.2.1 GlaS dataset

comprises 85 training images and 80 testing images for colorectal cancer segmentation. The images are 775x522 pixel patches from colon WSIs at 20X, each annotated with pixel-wise epithelial gland masks. The testing set is divided into subset A with 60 images and subset B with 20 images. We present results for the combined test set, comparing object-level dice coefficients and F1 scores to fully-supervised methods from previous studies [LLP22]. We also assessed our model’s performance across varying training data subsets (from 20% to 80%) and benchmarked it against leading semi-supervised learning methods.

Table 7.3 details the segmentation results using the full training dataset. Our approach, employing a frozen pretrained model as a static feature extractor with appended upsampling layers, delivered performance on par with sophisticated baselines trained from scratch such as FCDenseNet with a pyramid structure and multi-scale conditional GAN [LLP22]. Fine-tuning with LoRA further enhanced our model’s performance, closely approaching the state-of-the-art [GER19].

Table 7.4 presents results from experiments using varied training data subsets to test robustness under limited data conditions, including a few-shot learning scenario (with 20%

Table 7.3: Model performance on the GlaS challenge test set using the full training set.

Method	Object dice	F1 score
FCDenseNet [LLP22]	0.748	0.676
FCDenseNet + pyramid [LLP22]	0.870	0.860
Multi-scale conditional GAN [LLP22]	0.874	0.866
FCN-8 [LSD15]	0.781	0.763
DeepLab [CPK17]	0.833	0.813
Seg-Net [BKC17]	0.838	0.806
U-Net [RFB15]	0.868	0.841
DCAN [CQY16]	0.868	0.863
Rota-net [GER19]	0.902	0.896
Ours-static	0.872	0.862
Ours-LoRA	0.893	0.885

training data equating to only 17 images). All baselines are semi-supervised methods tested under an inductive learning setting, unless otherwise noted, where transductive learning treated images in the test set as unlabeled data for training. And the multi-scale conditional GAN further incorporated synthetic images for training. Unlike these methods, our model was trained solely on the training subset, without exposure to the test set or synthetic images. Results indicate that our approach enhances performance particularly in low-data regimes, suggesting that our pretrained model weights provide extra regularization and robust prior knowledge, stabilizing training and adaptation for downstream segmentation tasks, particularly in scenarios with limited annotations.

7.4.2.2 Prostate Gleason Grading Dataset

consists of 513 images from the archives of the Pathology Department at Cedars-Sinai Medical Center (IRB# Pro00029960), involving 40 patients across two subsets. Subset A in-

Table 7.4: Model performance on the GlaS challenge test set across varying training subsets. Results are based on inductive learning unless otherwise noted in parentheses.

Method	20%		40%		60%		80%	
	Dice	F1	Dice	F1	Dice	F1	Dice	F1
FCDenseNet [LLP22]	0.643	0.572	0.661	0.586	0.675	0.604	0.681	0.616
FCDenseNet + pyramid [LLP22]	0.651	0.570	0.655	0.601	0.789	0.716	0.818	0.825
Multi-scale conditional GAN [LLP22]	0.678	0.596	0.699	0.593	0.833	0.843	0.837	0.839
Multi-scale conditional GAN (transductive) [LLP22]	0.703	0.624	0.725	0.696	0.815	0.776	0.840	0.844
Ours-static	0.709	0.626	0.723	0.691	0.828	0.827	0.839	0.841
Ours-LoRA	0.727	0.693	0.813	0.764	0.842	0.841	0.859	0.852

cludes 224 images depicting stroma (ST), benign glands (BN), low-grade (LG), and high-grade (HG) cancers. Subset B contains 289 images primarily of dense high-grade tumors. The images are 1200x1200 pixel tiles from WSIs at 20X, manually annotated by expert pathologists. Following methodologies from previous work [LLP22], we implemented a 5-fold cross-validation strategy. We evaluated segmentation performance using individual Jaccard coefficients across four label classes, along with the mean Intersection Over Union (mIOU). To ensure a fair comparison with semi-supervised methods reported by [LLP22], we trained our model five times using randomly sampled training data at different ratios (from 20% to 80%) and reported average performance on a fixed test set.

Table 7.5 presents comparative model performance on the prostate test set using the full training dataset. Our pretrained model with LoRA exhibited superior performance, particularly in segmenting stroma regions, where it surpassed the best baseline by 5.4% in Jaccard score. This enhancement may be attributed to our pretrained dataset being a large-scale, heterogeneous collection covering multiple organs and disease types, which provides robust inductive biases for identifying common structures like stroma. Table 7.6 highlights the robust performance of our model across various training subsets, noting particularly significant gains in more constrained data scenarios: a 7.9% improvement over the state-of-the-art baseline with 20% of the training set.

Table 7.5: Model performance in segmenting prostate dataset images into "stroma" (BG), "benign" (BN), "low-grade" (LG), and "high-grade" (HG) categories.

Method	J_{BG}	J_{BN}	J_{LG}	J_{HG}	mIOU
FCDenseNet [LLP22]	0.566	0.492	0.614	0.524	0.549
Multi-scale conditional GAN [LLP22]	0.826	0.741	0.713	0.786	0.767
Handcrafted Features [GIM15]	0.595	0.352	0.495	N/A	0.481
Multi-Scale U-Net [LSH17]	0.824	0.721	0.587	0.784	0.729
FCN-8s [IML18]	N/A	N/A	N/A	N/A	0.759
Path R-CNN [LLS18]	0.831	0.839	0.715	0.797	0.796
Ours-static	0.835	0.792	0.731	0.814	0.793
Ours-LoRA	0.885	0.843	0.734	0.824	0.822

Table 7.6: mIOU on the prostate dataset across various training subsets. Results are based on inductive learning unless otherwise noted in parentheses.

Method	20%	40%	60%	80%
FCDenseNet [LLP22]	0.387 ± 0.037	0.420 ± 0.027	0.406 ± 0.025	0.492 ± 0.012
FCDenseNet + pyramid [LLP22]	0.437 ± 0.051	0.500 ± 0.039	0.581 ± 0.029	0.661 ± 0.017
Consistency-based [LLP22]	0.470 ± 0.076	0.499 ± 0.042	0.612 ± 0.070	0.658 ± 0.023
Consistency-based (transductive) [LLP22]	0.497 ± 0.056	0.531 ± 0.040	0.629 ± 0.037	0.667 ± 0.028
Multi-scale conditional GAN [LLP22]	0.527 ± 0.063	0.573 ± 0.045	0.632 ± 0.030	0.694 ± 0.021
Multi-scale conditional GAN (transductive) [LLP22]	0.577 ± 0.045	0.620 ± 0.028	0.664 ± 0.027	0.721 ± 0.013
Ours-static	0.618 ± 0.036	0.653 ± 0.024	0.684 ± 0.028	0.725 ± 0.011
Ours-LoRA	0.656 ± 0.054	0.689 ± 0.049	0.715 ± 0.038	0.752 ± 0.026

7.4.3 Whole Slide Image Classification

To evaluate our model’s capability to directly analyze WSIs at the gigapixel level, we utilized the CAMELYON16 dataset [BVV17], a widely-used public histopathology dataset designed for lymph node classification and metastasis detection. This dataset comprises 399 H&E stained WSIs of sentinel lymph node sections, categorized into 270 training slides (110 with

Table 7.7: Whole slide image classification results on Camelyon16

Method	ACC	AUC
CLAM-SB [LWC21]	0.877 ± 0.029	0.933 ± 0.002
CLAM-MB [LWC21]	0.894 ± 0.010	0.938 ± 0.005
TransMIL [SBC21]	0.905 ± 0.005	0.950 ± 0.005
DTFD-MIL [ZMZ22]	0.889 ± 0.007	0.941 ± 0.005
GTP [ZGG22]	0.883 ± 0.026	0.921 ± 0.026
CAMIL-L [FDB24]	0.910 ± 0.010	0.953 ± 0.002
CAMIL-G [FDB24]	0.891 ± 0.001	0.950 ± 0.009
CAMIL [FDB24]	0.917 ± 0.006	0.959 ± 0.001
Ours-LoRA	0.929 ± 0.008	0.964 ± 0.002
Resnet-50-mean	0.663 ± 0.010	0.625 ± 0.003
Resnet-50-max	0.691 ± 0.005	0.673 ± 0.002
Ours-static-mean	0.748 ± 0.009	0.774 ± 0.003
Ours-static-max	0.805 ± 0.004	0.813 ± 0.001

nodal metastases, 160 without) and 129 test slides (49 with nodal metastases, 80 without). These slides have been meticulously annotated by pathologists and were acquired at two different centers using scanners equipped with 20x and 40x objective lenses.

Given the relative ease of training fully supervised patch-level models with detailed tumor region annotations, we aim to evaluate our model’s capability to process WSI-level gigapixel images under weak supervision. Our model focuses on classifying WSIs using only slide-level labels. Specifically we employ the LoRA strategy to finetune pretrained model with dilated attention to enhance contextual understanding from long sequences of patch tokens. We then use max pooling to consolidate patch features into a single WSI representation, followed by a binary classification layer. We first establish baselines by extracting patch features with a frozen pretrained model, mean/max pooling the embeddings as the slide-level

representation, and evaluating with a KNN classifier. For comparison with state-of-the-art weakly supervised learning models, we adopted the evaluation settings from previous work [FDB24]. Specifically, we split the training set into 80% training and 20% validation segments in a 5-fold cross-validation scheme. We then report the model’s average performance on the CAMELYON16 challenge test set, measuring accuracy and AUROC.

Table 7.7 compares our method to several state-of-the-art baselines predominantly within the multiple instance learning (MIL) framework, which typically involves complex patch aggregation operations with more computation cost. Specifically, CLAM [LWC21] incorporates an attention-based pooling operator within the AB-MIL framework [ITW18] to enhance tile feature analysis, and TransMIL [SBC21] introduces a transformer-based aggregator. DTFT-MIL [ZMZ22] employs class activation maps to estimate instance positivity under AB-MIL, while GTP [ZGG22] combines a graph-based WSI representation with a vision transformer. CAMIL [FDB24] uses a context-aware MIL architecture with neighbor-constrained attention to manage tile dependencies. Our proposed method surpassed all baselines in both test accuracy and AUROC. Its magnification independence enables flexibility in experimenting with patches at various magnifications, accommodating different resolutions and computational complexities. Since our performance at 10X magnification is slightly better than at 20X, and importantly, opting for 10X leads to a shorter patch sequence length and requires less computation, we have chosen to report our results at 10X. In KNN evaluations with frozen model backbones, our model significantly outperformed a pretrained ResNet-50 on ImageNet, underscoring the advantages of our pathology domain-specific pretraining.

In WSI-level experiments, our model efficiently enhances contextual feature aggregation using dilated attention, achieving improved performance without increasing complexity or computational costs. Our pretrained VQGAN effectively compresses patches into discrete tokens, facilitating scalable Transformer pretraining and adaptation. Additionally, the scale-aware positional encoding module demonstrates its generalization ability in capturing spatial relationships among long patch token sequences.

7.5 Discussion

7.5.1 Histopathology Image Reconstruction and Synthesis

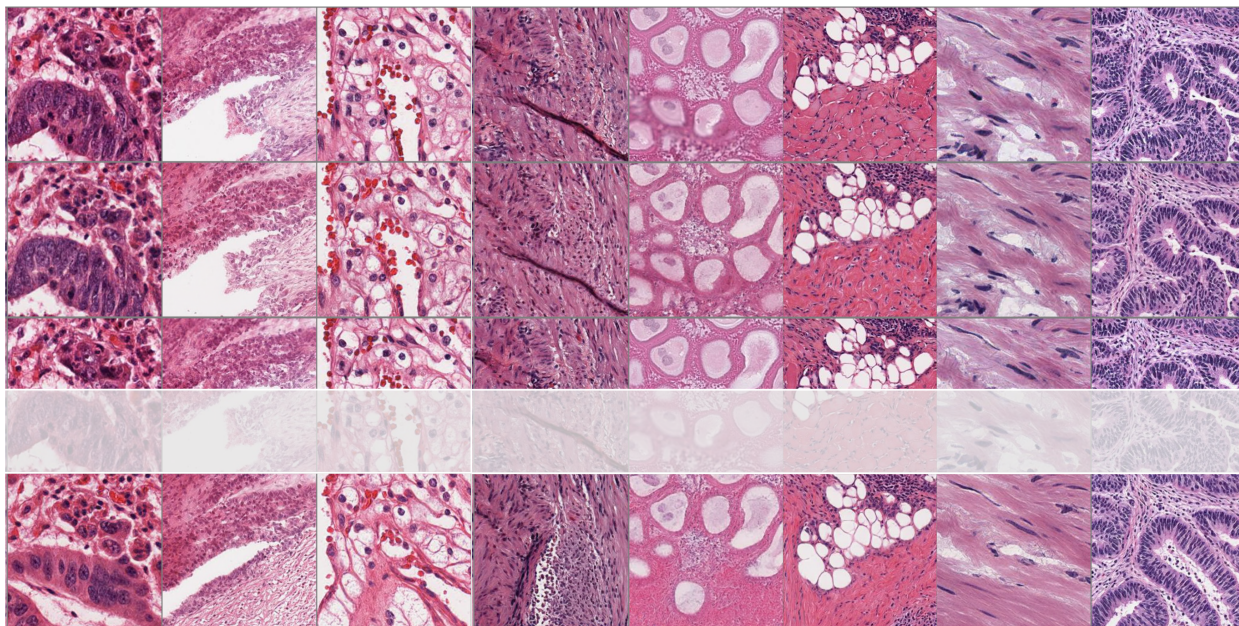


Figure 7.4: Images reconstructed by VQGAN and generated by latent Transformer. (1) First row: Original input images. (2) Second row: Images reconstructed from pretrained VQGAN using learned token codebook. (3) Third row: Source images where only the upper half is visible to the latent Transformer for image outpainting via autoregressive prediction of the next image token. (4) Fourth row: Images generated by pretrained latent Transformer by extending the third row source images.

In the first two rows of Fig. 7.4, we display reconstruction results from our pretrained VQGAN, highlighting the model’s ability to maintain excellent perceptual quality despite a high compression rate. The reconstructions accurately preserve both the global structure and detailed features such as color, texture patterns, and critical pathological elements like nuclei and stroma. Crucially, the high-quality reconstruction remains consistent across

various magnifications, demonstrating the efficacy of our pretraining strategy in enabling the VQGAN to learn a multi-magnification, perceptually rich codebook, which is vital for effective subsequent Transformer modeling.

Our pretrained Transformer adeptly handles class-unconditional image generation without requiring fine-tuning of the model parameters. The bottom half of Fig. 7.4 illustrates the results of image outpainting, where the model predicts subsequent image tokens to extend the source images. The generated samples not only incorporate realistic global structures but also detailed features that align with the hidden areas of the source images. Rather than merely duplicating visible regions, the generated areas offer plausible extrapolations based on the contextual information from the source images. This capability suggests that our Transformer has learned a robust representation for image composition modeling across various magnifications.

The Transformer architecture’s versatility highlights its potential for image synthesis tasks. In conditional settings, it can integrate condition information such as image-level class labels or semantic segmentation masks to enhance image distribution learning. Building on previous research that underscores the benefits of synthetic data in improving representation learning, particularly in data-limited scenarios ([LLP22]), our model can be adapted to a conditional generator. This adaptation would enable the production of synthetic data with tailored annotations to improve representation learning. Specifically, it would allow for the creation of synthetic annotations, including image-level labels and pixel mask annotations, which are crucial for enhancing downstream learning tasks such as classification and segmentation. Moreover, our approach could theoretically generate images of any ratio and size, providing the Transformer with a comprehensive contextual framework for image generation in both multi-resolution and multi-magnification settings.

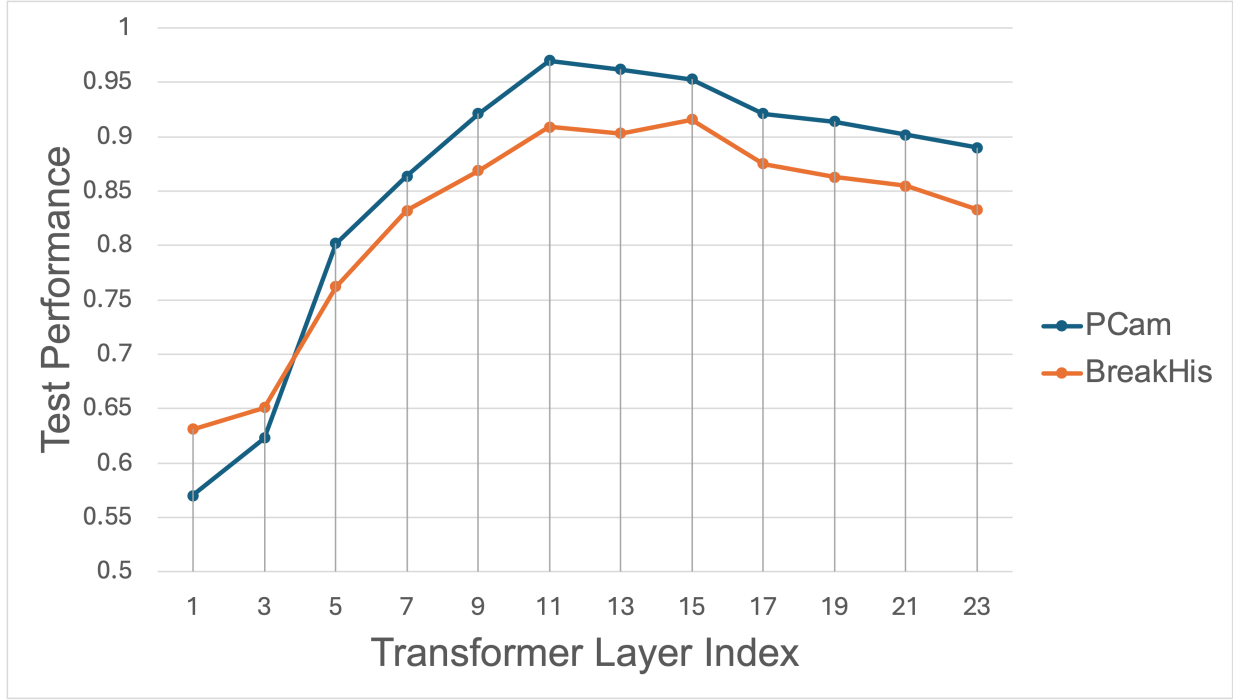


Figure 7.5: Accuracy of linear probing experiments using features extracted from different Transformer layers for patch-level image classification tasks.

7.5.2 Impact of Transformer Layer on Representation Quality

Fig. 7.5 illustrates the test accuracy of linear probing experiments using features extracted from various Transformer layers. We observe a rapid increase in linear-probing accuracy from the outputs of the initial Transformer layers, peaking at the middle layers (specifically, layer 11 for the PCam experiment and layer 15 for the BreakHis experiments). This pattern aligns with findings from previous research [CRC20, YLK21], which suggest that in the Transformer model, earlier layers function similarly to an encoder by converting image tokens into high-level semantic features, while later layers act more like a decoder, using these features to predict subsequent image tokens. Consequently, the final layers may not provide the most discriminative features for patch-level classification tasks, as the generative pretraining objective primarily focuses on reconstructing the original image tokens. For simplicity in downstream experiments, we consistently use the Transformer layer indexed at

13 for all other applications, including patch-level segmentation and whole slide image classification. Future studies could further explore the optimal Transformer layers for different downstream tasks.

7.6 Conclusion

In this study, we present a self-supervised learning framework for computational pathology that unifies image generation and representation learning. We propose efficient pretraining and finetuning strategies to address the limitations of existing SSL methods in modeling the gigapixel and hierarchical structures of WSIs. Our extensive experiments across various tissue types and magnification levels have shown that our model learns robust and generalizable representations for various downstream tasks. Moving forward, we aim to scale up the model and pretraining dataset size. Future research could explore the interpretation of the discrete visual codebook learned by the VQGAN and extend the latent Transformer to a conditional image generator. This would improve representation learning with limited data by utilizing synthetic labeled data. We anticipate that our framework will serve as a versatile backbone for scalable, data-efficient deep learning research in digital pathology.

CHAPTER 8

Conclusion and Future Work

This Chapter summarizes the main contributions of this dissertation and discusses potential future directions to investigate.

8.1 Summary of Contributions

This dissertation presents graph deep learning models for biomedical research across multiple scales. From a methodological perspective, our proposed methods address key graph learning tasks including link prediction, node classification, graph-level modeling, and effective model pretraining. From an applications perspective, we enhance heterogeneous data analysis for drug repurposing and EHR modeling. We also advance pathology image analysis through efficient graph representation learning. Furthermore, we introduce a self-supervised learning framework tailored for computational pathology, which is designed to facilitate rapid adaptation for future data-efficient deep learning research.

1. In Chapter 3, we propose a bipartite graph convolutional network for drug repurposing through heterogeneous information fusion. Our approach integrates multi-scale pharmaceutical information by constructing a unified multi-relational graph, comprising drug-protein, disease-protein, and protein-protein interactions (PPI). Notably, our model introduces protein nodes as a bridge for message passing among diverse biological domains, providing insights into utilizing PPI for improved drug repurposing assessment.

2. In Chapter 4, we propose a temporal graph learning framework that harnesses the rich information embedded within heterogeneous EHRs to enhance disease modeling. Our approach reconstructs sequential EHR data into temporal graphs, integrating time-aware information propagation with a heterogeneous graph attention mechanism. Our approach enables the capture of intricate dynamic patterns by constructing temporal graphs that interconnect diverse patient record sequences. Our research contributes to the advancement of clinical decision systems for depression prediction, offering a scalable solution to facilitate timely and personalized patient care.
3. In Chapters 5 and 6, we present graph deep learning approaches for histopathology image analysis across multiple magnification levels. We propose a hierarchical graph convolutional network framework for prostate cancer survival prediction using WSIs, modeling nuclei-level and patch-level graph structures in an integrative manner. Furthermore, we apply spatial graph modeling to enable accurate and efficient analysis of the tumor microenvironment in prostate cancer pathology. Our graph deep learning approach characterizes prostate tumor-associated stroma by integrating pathology image analysis from both whole-mount and biopsy slides. Our research offers a computational framework for exploring the field effect and cancer progression in prostate cancer, potentially contributing to improved prognostic models and targeted therapeutic strategies.
4. In Chapter 7, we propose a self-supervised learning framework for computational pathology that jointly addresses image generation and representation learning. We develop a domain-specific image tokenizer that encodes diverse pathological phenotypes through vector-quantized image modeling. Furthermore, we leverage a scale-aware GPT to learn foundational WSI representations that generalize robustly across multiple magnifications. Our approach also encompasses efficient fine-tuning strategies that adapt the pretrained model to downstream tasks, enhancing the development of data-efficient deep learning models suitable for diverse clinical applications.

8.2 Future Work

There are still many open research problems in graph deep learning for biomedicine that this dissertation doesn't cover.

8.2.1 Systematic Evaluation of Graph Model Explainability

For graph deep learning models to be effectively integrated into clinical practice, establishing trust with human users is crucial. Explainability—the ability of a model to justify its outcomes—is vital for building this trust, helping users understand the basis of predictions. This aspect of model transparency is particularly critical in healthcare, where grasping the rationale behind model decisions goes beyond conventional performance metrics. In the realm of deep learning, explainability has predominantly concentrated on generating input-dependent explanations and dissecting model behavior through various lenses, such as visual representations and analysis of attention value distributions. For instance, investigating the gradients or weights can reveal how sensitive the predictions are to changes in input features. However, the systematic exploration and visualization of explainability in graph-based deep learning remain underdeveloped compared to Euclidean domains.

Furthermore, the challenge of enhancing explainability with more "graph-oriented" approaches remains unresolved. One potential direction is developing methods to efficiently identify critical subgraph structures that contribute to model explainability, thereby elucidating the interconnections between graph entities. Such graph-centric insights could significantly advance biomedical research by providing deeper relational insights.

8.2.2 Graph Structure Learning and Dynamic Graph Modeling

A critical question that emerges is the origin of the graphs upon which graph networks operate. How do we determine the optimal graph structure? As discussed in this dissertation, one approach is to assume a fully connected graph structure among entities, utilizing a

Transformer model. However, this may not accurately represent the "true" relationships between entities. Many real-world graph structures are far more sparse than fully connected graphs, raising the question of how to effectively induce this sparsity. Developing methods to integrate graph embedding with structure learning presents a significant challenge and an exciting opportunity for future research.

Another pressing issue is how to dynamically modify graph structures during model computation. Dynamic graphs provide a more natural representation of many biomedical systems. For instance, in Chapter 4, we treated EHR data as a temporal graph, assigning timestamps to graph edges without altering the structure during training. Future work could expand this approach into a general graph-based pretraining framework that predicts the next graph edge under dynamic settings.

8.2.3 Multi-modal Self-supervised Learning Framework

The Transformer model offers a unified architecture for computer vision and natural language processing. In Chapter 7, we introduced a visual tokenizer tailored for pathology image analysis. Moving forward, we plan to leverage the extensive textual data from pathology reports and other imaging modalities, such as radiology. The next step involves expanding our self-supervised learning framework into a multi-modal architecture. This development will enable the seamless integration of diverse information sources, enhancing the reliability and robustness of diagnostic and treatment predictions.

REFERENCES

- [AGC07] Muhammed AY, Kwang-Il Goh, Michael E Cusick, Albert-Laszlo Barabasi, Marc Vidal, et al. “Drug–target network.” *Nature biotechnology*, **25**(10):1119–1127, 2007.
- [AKT20] Mohammed Adnan, Shivam Kalra, and Hamid R Tizhoosh. “Representation Learning of Histopathology Images using Graph Neural Networks.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 988–989, 2020.
- [AOE23] Ali Amirahmadi, Mattias Ohlsson, and Kobra Etminani. “Deep learning prediction models based on EHR trajectories: A systematic review.” *Journal of biomedical informatics*, p. 104430, 2023.
- [BDP21] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. “Beit: Bert pre-training of image transformers.” *arXiv preprint arXiv:2106.08254*, 2021.
- [BGL11] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. “Network medicine: a network-based approach to human disease.” *Nature reviews genetics*, **12**(1):56–68, 2011.
- [BHB18] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. “Relational inductive biases, deep learning, and graph networks.” *arXiv preprint arXiv:1806.01261*, 2018.
- [BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation.” *IEEE transactions on pattern analysis and machine intelligence*, **39**(12):2481–2495, 2017.
- [BKH16a] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization.” *arXiv preprint arXiv:1607.06450*, 2016.
- [BKH16b] Neslihan Bayramoglu, Juho Kannala, and Janne Heikkilä. “Deep learning for magnification independent breast cancer histopathology image classification.” In *2016 23rd International conference on pattern recognition (ICPR)*, pp. 2440–2445. IEEE, 2016.
- [BKT03] M Boueuf, J KovAkf, and O TALrvov. “Inhibition of adhesion breast cancer cells by anticoagulant drugs and cimetidine.” *Neoplasma*, **50**:2, 2003.
- [BLK16] MPG Broen, AFG Leentjens, S Köhler, ML Kuijf, WM McDonald, and IH Richard. “Trajectories of recovery in depressed Parkinson’s disease patients treated with paroxetine or venlafaxine.” *Parkinsonism & related disorders*, **23**:80–85, 2016.

- [BLV18] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. “Roto-translation covariant convolutional networks for medical image analysis.” In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pp. 440–448. Springer, 2018.
- [BMR20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. “Language models are few-shot learners.” *Advances in neural information processing systems*, **33**:1877–1901, 2020.
- [Bod04] Olivier Bodenreider. “The unified medical language system (UMLS): integrating biomedical terminology.” *Nucleic acids research*, **32**(suppl_1):D267–D270, 2004.
- [BP17] Adam S Brown and Chirag J Patel. “A standard database for drug repositioning.” *Scientific data*, **4**:170029, 2017.
- [BR12] David A Barron and David R Rowley. “The reactive stroma microenvironment and prostate cancer progression.” *Endocrine-related cancer*, **19**(6):R187–R204, 2012.
- [BTJ20] Francesco Bonollo, George N Thalmann, Marianna Kruithof-de Julio, and Sofia Karkampouna. “The role of cancer-associated fibroblasts in prostate cancer tumorigenesis.” *Cancers*, **12**(7):1887, 2020.
- [BVV17] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermesen, Quirine F Manson, Maschenka Balkenhol, et al. “Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer.” *Jama*, **318**(22):2199–2210, 2017.
- [CDH18] Feixiong Cheng, Rishi J Desai, Diane E Handy, Ruisheng Wang, Sebastian Schneeweiss, Albert-László Barabási, and Joseph Loscalzo. “Network-based approach to prediction and population-based validation of in silico drug repurposing.” *Nature communications*, **9**(1):1–12, 2018.
- [CGF14] Edward Chesney, Guy M Goodwin, and Seena Fazel. “Risks of all-cause and suicide mortality in mental disorders: a meta-review.” *World psychiatry*, **13**(2):153–160, 2014.
- [CKB19] Feixiong Cheng, István A Kovács, and Albert-László Barabási. “Network-based prediction of drug combinations.” *Nature communications*, **10**(1):1–11, 2019.
- [CKN20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations.” In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

- [CLL20] Sacheth Chandramouli, Patrick Leo, George Lee, Robin Elliott, Christine Davis, Guangjing Zhu, Pingfu Fu, Jonathan I Epstein, Robert Veltri, and Anant Madabhushi. “Computer Extracted Features from Initial H&E Tissue Biopsies Predict Disease Progression for Prostate Cancer Patients on Active Surveillance.” *Cancers*, **12**(9):2708, 2020.
- [CLS19] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. “Cluster-gen: An efficient algorithm for training deep and large graph convolutional networks.” In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 257–266, 2019.
- [CLW20] Richard J Chen, Ming Y Lu, Jingwen Wang, Drew FK Williamson, Scott J Rodig, Neal I Lindeman, and Faisal Mahmood. “Pathomic fusion: an integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis.” *IEEE Transactions on Medical Imaging*, 2020.
- [CPD21] Jie Chen, Aldo Pareja, Giacomo Domeniconi, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Timothy Kaler, Tao B Schardl, Charles E Leiserson, et al. “EVOLVING GRAPH CONVOLUTIONAL NETWORKS FOR DYNAMIC GRAPHS.”, August 19 2021. US Patent App. 16/790,682.
- [CPK17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.” *IEEE transactions on pattern analysis and machine intelligence*, **40**(4):834–848, 2017.
- [CQY16] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. “DCAN: deep contour-aware networks for accurate gland segmentation.” In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, 2016.
- [CR18] Yifan Chen and Maarten de Rijke. “A collective variational autoencoder for top-n recommendation with side information.” In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, pp. 3–9, 2018.
- [CRC20] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. “Generative pretraining from pixels.” In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020.
- [CWF17] Kelsy C Cotto, Alex H Wagner, Yang-Yang Feng, Susanna Kiwala, Adam C Coffman, Gregory Spies, Alex Wollam, Nicholas C Spies, Obi L Griffith, and Malachi Griffith. “DGIdb 3.0: a redesign and expansion of the drug–gene interaction database.” *Nucleic acids research*, **46**(D1):D1068–D1073, 2017.

- [CXL20] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. “Learning the graphical structure of electronic health records with graph convolutional transformer.” In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 606–613, 2020.
- [CZS17] Jianfei Chen, Jun Zhu, and Le Song. “Stochastic training of graph convolutional networks with variance reduction.” *arXiv preprint arXiv:1710.10568*, 2017.
- [DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering.” In *Advances in neural information processing systems*, pp. 3844–3852, 2016.
- [DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*, 2018.
- [DGJ16] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Benjamin L King, Roy McMorran, Jolene Wieggers, Thomas C Wieggers, and Carolyn J Mattingly. “The comparative toxicogenomics database: update 2017.” *Nucleic acids research*, **45**(D1):D972–D978, 2016.
- [DIA12] Sivanesan Dakshanamurthy, Naiem T Issa, Shahin Assefnia, Ashwini Seshasayee, Oakland J Peters, Subha Madhavan, Aykut Uren, Milton L Brown, and Stephen W Byers. “Predicting new indications for approved drugs using a proteochemometric method.” *Journal of medicinal chemistry*, **55**(15):6832–6848, 2012.
- [DKJ20] Cameron Davidson-Pilon, Jonas Kalderstam, Noah Jacobson, Ben Kuhn, Paul Zivich, Mike Williamson, Deepyaman Datta, Andrew Fiore-Gartland, Alex Parij, Daniel Wilson, et al. “CamDavidsonPilon/lifelines: v0. 24.15.” *Zenodo*, 2020.
- [DLL20] Kexin Ding, Qiao Liu, Edward Lee, Mu Zhou, Aidong Lu, and Shaoting Zhang. “Feature-Enhanced Graph Networks for Genetic Mutational Prediction Using Histopathological Images in Colon Cancer.” In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 294–304. Springer, 2020.
- [DM19] Christopher G Davey and Patrick D McGorry. “Early intervention for depression in young people: a blind spot in mental health care.” *The Lancet Psychiatry*, **6**(3):267–272, 2019.
- [DMD23] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. “Longnet: Scaling transformers to 1,000,000,000 tokens.” *arXiv preprint arXiv:2307.02486*, 2023.

- [DSR17] Andrea Diaz De Vivar, Mohammad Sayeeduddin, David Rowley, Antonio Cubilla, Brian Miles, Dov Kadmon, and Gustavo Ayala. “Histologic features of stromogenic carcinoma of the prostate (carcinomas with reactive stroma grade 3).” *Human pathology*, **63**:202–211, 2017.
- [EMP18] Babak Ehteshami Bejnordi, Maeve Mullooly, Ruth M Pfeiffer, Shaoqi Fan, Pamela M Vacek, Donald L Weaver, Sally Herschorn, Louise A Brinton, Bram van Ginneken, Nico Karssemeijer, et al. “Using deep convolutional neural networks to identify and classify tumor-associated stroma in diagnostic breast biopsies.” *Modern Pathology*, **31**(10):1502–1512, 2018.
- [ERO21] Patrick Esser, Robin Rombach, and Bjorn Ommer. “Taming transformers for high-resolution image synthesis.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.
- [EVS16] Jonathan I Epstein, Michael J Zelefsky, Daniel D Sjoberg, Joel B Nelson, Lars Egevad, Cristina Magi-Galluzzi, Andrew J Vickers, Anil V Parwani, Victor E Reuter, Samson W Fine, et al. “A contemporary prostate cancer grading system: a validated alternative to the Gleason score.” *European urology*, **69**(3):428–435, 2016.
- [FBS17] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. “Protein interface prediction using graph convolutional networks.” In *Advances in Neural Information Processing Systems*, pp. 6530–6539, 2017.
- [FDB24] Olga Fourkioti, Matt De Vries, and Chris Bakal. “CAMIL: Context-Aware Multiple Instance Learning for Cancer Detection and Subtyping in Whole Slide Images.” In *The Twelfth International Conference on Learning Representations*, 2024.
- [FDC18] Qingyuan Feng, Evgenia Dueva, Artem Cherkasov, and Martin Ester. “Padme: A deep learning-based framework for drug-target interaction prediction.” *arXiv preprint arXiv:1807.09741*, 2018.
- [FLC16] A Spanhol Fabio, S Oliveira Luiz, Petitjean Caroline, and Heutte Laurent. “Breast cancer histopathological image classification using convolutional neural networks.” In *Conference: International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2016.
- [FMN17] Susan H Fox, Leonard Verhagen Metman, John G Nutt, Matthew Brodsky, Stewart A Factor, Anthony E Lang, Laura E Pope, Nadine Knowles, and João Siffert. “Trial of dextromethorphan/quinidine to treat levodopa-induced dyskinesia in Parkinson’s disease.” *Movement Disorders*, **32**(6):893–903, 2017.

- [FSK19] Michael Fralick, Chana A Sacks, and Aaron S Kesselheim. “Assessment of use of combined dextromethorphan and quinidine in patients with dementia or Parkinson disease after US Food and Drug Administration approval for pseudobulbar affect.” *JAMA internal medicine*, **179**(2):224–230, 2019.
- [GB10] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks.” In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.
- [GB17] Vibha Gupta and Arnav Bhavsar. “Breast cancer histopathological image classification: is magnification important?” In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 17–24, 2017.
- [GB18] Vibha Gupta and Arnav Bhavsar. “Sequential modeling of deep features for breast cancer histopathological image classification.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2254–2261, 2018.
- [GER19] Simon Graham, David Epstein, and Nasir Rajpoot. “Rota-net: Rotation equivariant network for simultaneous gland and lumen segmentation in colon histology images.” In *Digital Pathology: 15th European Congress, ECDP 2019, Warwick, UK, April 10–13, 2019, Proceedings 15*, pp. 109–116. Springer, 2019.
- [GER20] Simon Graham, David Epstein, and Nasir Rajpoot. “Dense steerable filter cnns for exploiting rotational symmetry in histology images.” *IEEE Transactions on Medical Imaging*, **39**(12):4124–4136, 2020.
- [GFL18] Kyle Yingkai Gao, Achille Fokoue, Heng Luo, Arun Iyengar, Sanjoy Dey, and Ping Zhang. “Interpretable Drug Target Prediction Using Deep Neural Representation.” In *IJCAI*, pp. 3371–3377, 2018.
- [GH10] Michael Gutmann and Aapo Hyvärinen. “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.” In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.
- [GIM15] Arkadiusz Gertych, Nathan Ing, Zhaoxuan Ma, Thomas J Fuchs, Sadri Salman, Sambit Mohanty, Sanica Bhele, Adriana Velásquez-Vacca, Mahul B Amin, and Beatrice S Knudsen. “Machine learning approaches to analyze histological images of tissues from radical prostatectomies.” *Computerized Medical Imaging and Graphics*, **46**:197–208, 2015.
- [GL17] Jeremy A Greene and Joseph Loscalzo. “Putting the Patient Back Together-Social Medicine, Network Medicine, and the Limits of Reductionism.” *The New England journal of medicine*, **377**(25):2493, 2017.

- [GMV16] Emre Guney, Jörg Menche, Marc Vidal, and Albert-László Barábasi. “Network-based in silico drug efficacy screening.” *Nature communications*, **7**:10331, 2016.
- [GSR11] Assaf Gottlieb, Gideon Y Stein, Eytan Ruppin, and Roded Sharan. “PREDICT: a method for inferring novel drug indications with application to personalized medicine.” *Molecular systems biology*, **7**(1), 2011.
- [HCX22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. “Masked autoencoders are scalable vision learners.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.
- [HFL18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. “Learning deep representations by mutual information estimation and maximization.” *arXiv preprint arXiv:1808.06670*, 2018.
- [HFW20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.
- [HGD17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn.” In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models.” *Advances in neural information processing systems*, **33**:6840–6851, 2020.
- [HJG20] Wenchao Han, Carol Johnson, Mena Gaed, José A Gómez, Madeleine Moussa, Joseph L Chin, Stephen Pautler, Glenn S Bauman, and Aaron D Ward. “Histologic tissue components provide major cues for machine learning-based prostate cancer detection and grading on prostatectomy specimens.” *Scientific reports*, **10**(1):9911, 2020.
- [HJW19] Zhen Han, Jindong Jiang, Yuyi Wang, Yunpu Ma, and Volker Tresp. “The graph hawkes network for reasoning on temporal knowledge graphs.” In *Learning with Temporal Point Processes Workshop at at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019) NeurIPS 2019*, 2019.
- [HLG19] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. “Strategies for pre-training graph neural networks.” *arXiv preprint arXiv:1905.12265*, 2019.

- [HLI14] Sandy H Huang, Paea LePendur, Srinivasan V Iyer, Ming Tai-Seale, David Carrell, and Nigam H Shah. “Toward personalizing treatment for depression: predicting diagnosis and severity.” *Journal of the American Medical Informatics Association*, **21**(6):1069–1075, 2014.
- [HLV17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely connected convolutional networks.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [HSA05] Ada Hamosh, Alan F Scott, Joanna S Amberger, Carol A Bocchini, and Victor A McKusick. “Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders.” *Nucleic acids research*, **33**(suppl.1):D514–D517, 2005.
- [HSW21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. “Lora: Low-rank adaptation of large language models.” *arXiv preprint arXiv:2106.09685*, 2021.
- [HWC01] Simon W Hayward, Yuzhuo Wang, Mei Cao, Yun Kit Hom, Baohui Zhang, Gary D Grossfeld, Daniel Sudilovsky, and Gerald R Cunha. “Malignant transformation in a nontumorigenic human prostatic epithelial cell line.” *Cancer research*, **61**(22):8135–8142, 2001.
- [HWZ17] Zhongyi Han, Benzheng Wei, Yuanjie Zheng, Yilong Yin, Kejian Li, and Shuo Li. “Breast cancer multi-classification from histopathological images with structured deep learning model.” *Scientific reports*, **7**(1):4172, 2017.
- [HXR19] Chaoyang He, Tian Xie, Yu Rong, Wenbing Huang, Yanfang Li, Junzhou Huang, Xiang Ren, and Cyrus Shahabi. “Bipartite Graph Neural Networks for Efficient Node Representation Learning.” *arXiv preprint arXiv:1906.11994*, 2019.
- [HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs.” In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.
- [HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [IML18] Nathan Ing, Zhaoxuan Ma, Jiayun Li, Hootan Salemi, Corey Arnold, Beatrice S Knudsen, and Arkadiusz Gertych. “Semantic segmentation for prostate cancer grading by convolutional neural networks.” In *Medical Imaging 2018: Digital Pathology*, volume 10581, pp. 343–355. SPIE, 2018.

- [ITW18] Maximilian Ilse, Jakub Tomczak, and Max Welling. “Attention-based deep multiple instance learning.” In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.
- [IVA22] Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. “Learning with neighbor consistency for noisy labels.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4672–4681, 2022.
- [Jin15] Haomiao Jin. “Development of a clinical forecasting model to predict comorbid depression among diabetes patients and an application in depression screening policy making.” *Preventing chronic disease*, **12**, 2015.
- [JJQ19] Woojeong Jin, He Jiang, Meng Qu, Tong Chen, Changlin Zhang, Pedro A. Szekely, and Xiang Ren. “Recurrent Event Network : Global Structure Inference Over Temporal Knowledge Graph.” *arXiv: Learning*, 2019.
- [JKK19] Won Sik Jang, Dong Hoon Koh, Jongchan Kim, Jong Soo Lee, Doo Yong Chung, Won Sik Ham, Koon Ho Rha, and Young Deuk Choi. “The prognostic impact of downgrading and upgrading from biopsy to radical prostatectomy among men with Gleason score 7 prostate cancer.” *The Prostate*, **79**(16):1805–1810, 2019.
- [KAT19] Boris Knyazev, Carolyn Augusta, and Graham W Taylor. “Learning temporal attention in dynamic graphs with bilinear interactions.” *arXiv preprint arXiv:1909.10367*, 2019.
- [KB14] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980*, 2014.
- [KCC18] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. “PubChem 2019 update: improved access to chemical data.” *Nucleic acids research*, **47**(D1):D1102–D1109, 2018.
- [KGE19] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupert, and Marcus Brubaker. “Time2vec: Learning a vector representation of time.” *arXiv preprint arXiv:1907.05321*, 2019.
- [KMB16] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. “Molecular graph convolutions: moving beyond fingerprints.” *Journal of computer-aided molecular design*, **30**(8):595–608, 2016.
- [KPD16] Panagiota I Kontou, Athanasia Pavlopoulou, Niki L Dimou, Georgios A Pavlopoulos, and Pantelis G Bagos. “Network analysis of genes and their association with diseases.” *Gene*, **590**(1):68–78, 2016.

- [KPP20] Roger S Kirby, Manish I Patel, and Darren MC Poon. *Fast Facts: Prostate Cancer: If, when and how to intervene*. Karger Medical and Scientific Publishers, 2020.
- [Kro21] Kurt Kroenke. “PHQ-9: global uptake of a depression scale.” *World Psychiatry*, **20**(1):135, 2021.
- [KSC16] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. “Deep survival: A deep cox proportional hazards network.” *stat*, **1050**(2), 2016.
- [KVA19] Neeraj Kumar, Ruchika Verma, Deepak Anand, Yanning Zhou, Omer Fahri Onder, Efstratios Tsougenis, Hao Chen, Pheng-Ann Heng, Jiahui Li, Zhiqiang Hu, et al. “A multi-organ nucleus segmentation challenge.” *IEEE transactions on medical imaging*, **39**(5):1380–1391, 2019.
- [KW13] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes.” *arXiv preprint arXiv:1312.6114*, 2013.
- [KW16] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks.” *arXiv preprint arXiv:1609.02907*, 2016.
- [KWH20] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. “Pointrend: Image segmentation as rendering.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9799–9808, 2020.
- [LBP21] Maxime W Lafarge, Erik J Bekkers, Josien PW Pluim, Remco Duits, and Mitko Veta. “Roto-translation equivariant convolutional networks: Application to histopathology image analysis.” *Medical Image Analysis*, **68**:101849, 2021.
- [LCM23] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. “Mage: Masked generative encoder to unify representation learning and image synthesis.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2142–2152, 2023.
- [LCT19] Zehuan Liao, Damien Chua, and Nguan Soon Tan. “Reactive oxygen species: a volatile driver of field cancerization and metastasis.” *Molecular cancer*, **18**:1–10, 2019.
- [LHC19] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. “Predicting path failure in time-evolving graphs.” In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1279–1289, 2019.

- [LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. “Deeper insights into graph convolutional networks for semi-supervised learning.” In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [LHZ22] Michelle M Li, Kexin Huang, and Marinka Zitnik. “Graph representation learning in biomedicine and healthcare.” *Nature Biomedical Engineering*, **6**(12):1353–1369, 2022.
- [Lip00] Carolyn E Lipscomb. “Medical subject headings (MeSH).” *Bulletin of the Medical Library Association*, **88**(3):265, 2000.
- [LJE19] Patrick Leo, Andrew Janowczyk, Robin Elliott, Nafiseh Janaki, Rakesh Shiradkar, Xavier Farrè, Kosj Yamoah, Timothy Rebbeck, Natalie Shih, Francesca Khani, et al. “Computerized histomorphometric features of glandular architecture predict risk of biochemical recurrence following radical prostatectomy: A multisite study.”, 2019.
- [LKC18] Katarzyna Lubecka, Agnieszka Kaufman-Szymczyk, Barbara Cebula-Obrzut, Piotr Smolewski, Janusz Szemraj, and Krystyna Fabianowska-Majewska. “Novel clofarabine-based combinations with polyphenols epigenetically reactivate retinoic acid receptor beta, inhibit cell growth, and induce apoptosis of breast cancer cells.” *International journal of molecular sciences*, **19**(12):3970, 2018.
- [LKS14] Katarzyna Lubecka-Pietruszewska, Agnieszka Kaufman-Szymczyk, Barbara Stefanska, Barbara Cebula-Obrzut, Piotr Smolewski, and Krystyna Fabianowska-Majewska. “Clofarabine, a novel adenosine analogue, reactivates DNA methylation-silenced tumour suppressor genes and inhibits cell growth in breast cancer cells.” *European journal of pharmacology*, **723**:276–287, 2014.
- [LL12] Jiao Li and Zhiyong Lu. “A new method for computational drug repositioning using drug pairwise similarity.” In *2012 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 1–4. IEEE, 2012.
- [LLH18] Jianfang Liu, Tara Lichtenberg, Katherine A Hoadley, Laila M Poisson, Alexander J Lazar, Andrew D Cherniack, Albert J Kovatich, Christopher C Benz, Douglas A Levine, Adrian V Lee, et al. “An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics.” *Cell*, **173**(2):400–416, 2018.
- [LLK19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. “Self-attention graph pooling.” In *International Conference on Machine Learning*, pp. 3734–3743. PMLR, 2019.
- [LLP22] Wenyuan Li, Jiayun Li, Jennifer Polson, Zichen Wang, William Speier, and Corey Arnold. “High resolution histopathology image generation and segmentation through adversarial training.” *Medical Image Analysis*, **75**:102251, 2022.

- [LLS18] Wenyuan Li, Jiayun Li, Karthik V Sarma, King Chung Ho, Shiwen Shen, Beatrice S Knudsen, Arkadiusz Gertych, and Corey W Arnold. “Path R-CNN for prostate cancer diagnosis and gleason grading of histological images.” *IEEE transactions on medical imaging*, **38**(4):945–954, 2018.
- [LLS21] Jiayun Li, Wenyuan Li, Anthony Sisk, Huihui Ye, W Dean Wallace, William Speier, and Corey W Arnold. “A Multi-resolution Model for Histopathology Image Classification and Localization with Multiple Instance Learning.” *Computers in Biology and Medicine*, p. 104253, 2021.
- [LMU18] Xiaowen Liu, Sophia Montissol, Amy Uber, Sarah Ganley, Anne V GrosseStreuer, Katherine Berg, Stanley Heydrick, and Michael W Donnino. “The effects of thiamine on breast cancer cells.” *Molecules*, **23**(6):1464, 2018.
- [LN18] Christine Levesque and Peter S Nelson. “Cellular constituents of the prostate stroma: Key contributors to prostate cancer progression and therapy resistance.” *Cold Spring Harbor perspectives in medicine*, **8**(8):a030510, 2018.
- [Loo13] Lynn H Loomis. *Introduction to abstract harmonic analysis*. Courier Corporation, 2013.
- [LRW18] Cheng Lu, David Romo-Bucheli, Xiangxue Wang, Andrew Janowczyk, Shridar Ganesan, Hannah Gilmore, David Rimm, and Anant Madabhushi. “Nuclear shape and orientation features from H&E images predict survival in early-stage estrogen receptor-positive breast cancers.” *Laboratory Investigation*, **98**(11):1438–1448, 2018.
- [LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- [LSH17] Jiayun Li, Karthik V Sarma, King Chung Ho, Arkadiusz Gertych, Beatrice S Knudsen, and Corey W Arnold. “A multi-scale u-net for semantic segmentation of histological images from radical prostatectomies.” In *AMIA Annual Symposium Proceedings*, volume 2017, p. 1140. American Medical Informatics Association, 2017.
- [LTB15] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. “Gated graph sequence neural networks.” *arXiv preprint arXiv:1511.05493*, 2015.
- [LTL18] Grace Y Lim, Wilson W Tam, Yanxia Lu, Cyrus S Ho, Melvyn W Zhang, and Roger C Ho. “Prevalence of depression in the community from 30 countries between 1994 and 2014.” *Scientific reports*, **8**(1):2861, 2018.

- [LWC21] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. “Data-efficient and weakly supervised computational pathology on whole-slide images.” *Nature biomedical engineering*, **5**(6):555–570, 2021.
- [LWL16] Huimin Luo, Jianxin Wang, Min Li, Junwei Luo, Xiaoqing Peng, Fang-Xiang Wu, and Yi Pan. “Drug repositioning based on comprehensive similarity measures and Bi-Random walk algorithm.” *Bioinformatics*, **32**(17):2664–2671, 2016.
- [LWL22] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. “A survey of transformers.” *AI open*, **3**:111–132, 2022.
- [LYX20] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. “Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records.” In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 647–656, 2020.
- [LYZ18] Ruoyu Li, Jiawen Yao, Xinliang Zhu, Yeqing Li, and Junzhou Huang. “Graph CNN for survival analysis on whole slide pathological images.” In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 174–182. Springer, 2018.
- [LZC15] Jiao Li, Si Zheng, Bin Chen, Atul J Butte, S Joshua Swamidass, and Zhiyong Lu. “A survey of current trends in computational drug repositioning.” *Briefings in bioinformatics*, **17**(1):2–12, 2015.
- [LZW23] Siyuan Li, Luyuan Zhang, Zedong Wang, Di Wu, Lirong Wu, Zicheng Liu, Jun Xia, Cheng Tan, Yang Liu, Baigui Sun, et al. “Masked Modeling for Self-supervised Representation Learning on Vision and Beyond.” *arXiv preprint arXiv:2401.00897*, 2023.
- [MBB20] Ewan KA Millar, Lois H Browne, Julia Beretov, Kirsty Lee, Jodi Lynch, Alexander Swarbrick, and Peter H Graham. “Tumour stroma ratio assessment using digital image analysis predicts survival in triple negative and luminal breast cancer.” *Cancers*, **12**(12):3749, 2020.
- [MCZ17] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. “Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks.” In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1903–1911, 2017.
- [MFH18] Samuel WD Merriel, Garth Funston, and Willie Hamilton. “Prostate cancer in primary care.” *Advances in therapy*, **35**:1285–1294, 2018.

- [MHY06] Patrick D McGorry, Ian B Hickie, Alison R Yung, Christos Pantelis, and Henry J Jackson. “Clinical staging of psychiatric disorders: a heuristic framework for choosing earlier, safer and more effective interventions.” *Australian & New Zealand Journal of Psychiatry*, **40**(8):616–622, 2006.
- [MLM12] Rodolfo Montironi, Antonio Lopez Beltran, Roberta Mazzucchelli, Liang Cheng, and Marina Scarpelli. “Handling of radical prostatectomy specimens: total embedding with large-format histology.” *International journal of breast cancer*, **2012**, 2012.
- [MRM20] Franco Manessi, Alessandro Rozza, and Mario Manzo. “Dynamic graph convolutional networks.” *Pattern Recognition*, **97**:107000, 2020.
- [MRO18] Patrick D McGorry, Aswin Ratheesh, and Brian O’Donoghue. “Early intervention—an implementation challenge for 21st century mental health care.” *JAMA psychiatry*, **75**(6):545–546, 2018.
- [MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality.” In *Advances in neural information processing systems*, pp. 3111–3119, 2013.
- [MSK15] Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. “Uncovering disease-disease relationships through the incomplete interactome.” *Science*, **347**(6224):1257601, 2015.
- [MSO20] Yiwen Meng, William Speier, Michael Ong, and Corey W Arnold. “HCET: hierarchical clinical embedding with topic modeling on electronic health records for predicting future depression.” *IEEE journal of biomedical and health informatics*, **25**(4):1265–1272, 2020.
- [MSO21] Yiwen Meng, William Speier, Michael K Ong, and Corey W Arnold. “Bidirectional representation learning from transformers using multimodal electronic health record data to predict depression.” *IEEE journal of biomedical and health informatics*, **25**(8):3121–3129, 2021.
- [MV13] Patrick McGorry and Jim Van Os. “Redeeming diagnosis in psychiatry: timing versus specificity.” *The Lancet*, **381**(9863):343–345, 2013.
- [MVK17] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. “Rotation equivariant vector field networks.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5048–5057, 2017.
- [MVR09] Alex J Mitchell, Amol Vaze, and Sanjay Rao. “Clinical diagnosis of depression in primary care: a meta-analysis.” *The Lancet*, **374**(9690):609–619, 2009.

- [MXZ18] Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. “Drug similarity integration through attentive multi-view graph auto-encoders.” *arXiv preprint arXiv:1804.10850*, 2018.
- [Nas18] Marcel Nassar. “Hierarchical bipartite graph convolution networks.” *arXiv preprint arXiv:1812.03813*, 2018.
- [NK12] Xia Ning and George Karypis. “Sparse linear methods with side information for top-n recommendations.” In *Proceedings of the sixth ACM conference on Recommender systems*, pp. 155–162, 2012.
- [NSG10] Rishi Nayyar, Prabhjot Singh, Narmada P Gupta, Ashok K Hemal, Prem N Dogra, Amlesh Seth, and Rajeev Kumar. “Upgrading of Gleason score on radical prostatectomy specimen compared to the pre-operative needle core biopsy: An Indian experience.” *Indian journal of urology: IJU: journal of the Urological Society of India*, **26**(1):56, 2010.
- [NZM13] Francesco Napolitano, Yan Zhao, Vânia M Moreira, Roberto Tagliaferri, Juha Kere, Mauro D’Amato, and Dario Greco. “Drug repositioning: a machine-learning approach through data integration.” *Journal of cheminformatics*, **5**(1):30, 2013.
- [OAA21] Diego Ortego, Eric Arazo, Paul Albert, Noel E O’Connor, and Kevin McGuinness. “Multi-objective interpolation training for robustness to label noise.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6606–6615, 2021.
- [OKE16] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. “Conditional image generation with pixelcnn decoders.” *Advances in neural information processing systems*, **29**, 2016.
- [PAS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations.” In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.
- [PBQ16] Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Furlong. “DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants.” *Nucleic acids research*, p. gkw943, 2016.
- [PDB22a] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. “BEiT v2: Masked Image Modeling with Vector-Quantized Visual Tokenizers.” *ArXiv*, **abs/2208.06366**, 2022.

- [PDB22b] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. “A unified view of masked image modeling.” *arXiv preprint arXiv:2210.10615*, 2022.
- [PKP18] Georgios A Pavlopoulos, Panagiota I Kontou, Athanasia Pavlopoulou, Costas Bouyioukos, Evripides Markou, and Pantelis G Bagos. “Bipartite graphs in systems biology and medicine: a survey of methods and applications.” *Gigascience*, **7**(4):giy014, 2018.
- [QCD20] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. “Gcc: Graph contrastive coding for graph neural network pre-training.” In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1150–1160, 2020.
- [RBL22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-resolution image synthesis with latent diffusion models.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- [RCF20] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. “Temporal graph networks for deep learning on dynamic graphs.” *arXiv preprint arXiv:2006.10637*, 2020.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation.” In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.
- [RGD22] Samuel Ruder, Yan Gao, Yi Ding, Ping Bu, Brian Miles, Angelo De Marzo, Thomas Wheeler, Jesse K McKenney, Heidi Auman, Ladan Fazli, et al. “Development and validation of a quantitative reactive stroma biomarker (qRS) for prostate cancer prognosis.” *Human pathology*, **122**:84–91, 2022.
- [RGS21] Jose F Rodrigues-Jr, Marco A Gutierrez, Gabriel Spadon, Bruno Brandoli, and Sihem Amer-Yahia. “LIG-Doctor: Efficient patient trajectory prediction using bidirectional minimal gated-recurrent networks.” *Information Sciences*, **545**:813–827, 2021.
- [RMC15] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks.” *arXiv preprint arXiv:1511.06434*, 2015.
- [RNR16] Charlotte L Rae, Cristina Nombela, Patricia Vázquez Rodríguez, Zheng Ye, Laura E Hughes, P Simon Jones, Timothy Ham, Timothy Rittman, Ian Coyle-Gilchrist, Ralf Regenthal, et al. “Atomoxetine restores the response inhibition network in Parkinson’s disease.” *Brain*, **139**(8):2235–2248, 2016.

- [RNS18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. “Improving language understanding by generative pre-training.” 2018.
- [ROC18] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. “Scalable and accurate deep learning with electronic health records.” *NPJ digital medicine*, **1**(1):1–10, 2018.
- [RWC19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. “Language models are unsupervised multitask learners.” *OpenAI blog*, **1**(8):9, 2019.
- [RWZ23] Sucheng Ren, Zeyu Wang, Hongru Zhu, Junfei Xiao, Alan Yuille, and Cihang Xie. “Rejuvenating image-GPT as Strong Visual Representation Learners.” *ArXiv*, **abs/2312.02147**, 2023.
- [SAB12] Philippe Sanseau, Pankaj Agarwal, Michael R Barnes, Tomi Pastinen, J Brent Richards, Lon R Cardon, and Vincent Mooser. “Use of genome-wide association studies for drug repositioning.” *Nature biotechnology*, **30**(4):317, 2012.
- [SBA16] Duncan E Scott, Andrew R Bayly, Chris Abell, and John Skidmore. “Small molecules, big targets: drug discovery faces the protein–protein interaction challenge.” *Nature Reviews Drug Discovery*, **15**(8):533, 2016.
- [SBC21] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. “Transmil: Transformer based correlated multiple instance learning for whole slide image classification.” *Advances in neural information processing systems*, **34**:2136–2147, 2021.
- [SCH17a] Ying Shi, Tong Cao, Hua Huang, Chaoqun Lian, Ying Yang, Zhiwei Wang, Jia Ma, and Jun Xia. “Arsenic trioxide inhibits cell growth and motility via up-regulation of let-7a in breast cancer cells.” *Cell Cycle*, **16**(24):2396–2403, 2017.
- [SCH17b] Yang Song, Hang Chang, Heng Huang, and Weidong Cai. “Supervised intra-embedding of fisher vectors for histopathology image classification.” In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20*, pp. 99–106. Springer, 2017.
- [SGM21] Joakim Skardinga, Bogdan Gabrys, and Katarzyna Musial. “Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey.” *IEEE Access*, 2021.
- [SGT08] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The graph neural network model.” *IEEE Transactions on Neural Networks*, **20**(1):61–80, 2008.

- [SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting.” *The journal of machine learning research*, **15**(1):1929–1958, 2014.
- [SK11] Karen L Smarr and Autumn L Keefer. “Measures of depression and depressive symptoms: Beck depression Inventory-II (BDI-II), center for epidemiologic studies depression scale (CES-D), geriatric depression scale (GDS), hospital anxiety and depression scale (HADS), and patient health Questionnaire-9 (PHQ-9).” *Arthritis care & research*, **63**(S11):S454–S466, 2011.
- [SKB18] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. “Modeling relational data with graph convolutional networks.” In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pp. 593–607. Springer, 2018.
- [SMJ19] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. “Cancer statistics, 2019.” *CA: a cancer journal for clinicians*, **69**(1):7–34, 2019.
- [SOP15] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. “A dataset for breast cancer histopathological image classification.” *Ieee transactions on biomedical engineering*, **63**(7):1455–1462, 2015.
- [SPC17] Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. “Gland segmentation in colon histology images: The glas challenge contest.” *Medical image analysis*, **35**:489–502, 2017.
- [SWG20] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. “Dysat: Deep neural representation learning on dynamic graphs via self-attention networks.” In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 519–527, 2020.
- [SZC17] Yang Song, Ju Jia Zou, Hang Chang, and Weidong Cai. “Adapting fisher vectors for histopathology image classification.” In *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*, pp. 600–603. IEEE, 2017.
- [Tan57] TT Tanimoto. “IBM Internal Report 17th.”, 1957.
- [TFB19] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. “Dyrep: Learning representations over dynamic graphs.” In *International conference on learning representations*, 2019.
- [UHK16] Oleg Ursu, Jayme Holmes, Jeffrey Knockel, Cristian G Bologa, Jeremy J Yang, Stephen L Mathias, Stuart J Nelson, and Tudor I Oprea. “DrugCentral: online drug compendium.” *Nucleic acids research*, p. gkw993, 2016.

- [VCC17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. “Graph attention networks.” *arXiv preprint arXiv:1710.10903*, 2017.
- [Vel23] Petar Veličković. “Everything is connected: Graph neural networks.” *Current Opinion in Structural Biology*, **79**:102538, 2023.
- [VFP17] Joost JM Van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts. “Computational radiomics system to decode the radiographic phenotype.” *Cancer research*, **77**(21):e104–e107, 2017.
- [VLW18] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. “Rotation equivariant CNNs for digital pathology.” In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, pp. 210–218. Springer, 2018.
- [VSC20] Rita L Vaz, Sara Sousa, Diana Chapela, Herma C van der Linde, Rob Willemssen, Ana D Correia, Tiago F Outeiro, and Nuno D Afonso. “Identification of antiparkinsonian drugs in the 6-hydroxydopamine zebrafish model.” *Pharmacology Biochemistry and Behavior*, **189**:172828, 2020.
- [VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” In *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [VV17] Aaron Van Den Oord, Oriol Vinyals, et al. “Neural discrete representation learning.” *Advances in neural information processing systems*, **30**, 2017.
- [WCC20] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. “TeMP: temporal message passing for temporal knowledge graph completion.” *arXiv preprint arXiv:2010.03526*, 2020.
- [WCL20] Jingwen Wang, Richard J Chen, Ming Y Lu, Alexander Baras, and Faisal Mahmood. “Weakly supervised prostate tma classification via graph convolutional networks.” In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 239–243. IEEE, 2020.
- [WDP07] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. “A new method to measure the semantic similarity of GO terms.” *Bioinformatics*, **23**(10):1274–1281, 2007.
- [Wei88] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.” *Journal of chemical information and computer sciences*, **28**(1):31–36, 1988.

- [WGT17] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. “Harmonic networks: Deep translation and rotation equivariance.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5028–5037, 2017.
- [WHS18] Maurice Weiler, Fred A Hamprecht, and Martin Storath. “Learning steerable filters for rotation equivariant cnns.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 849–858, 2018.
- [WMX23] Wenhui Wang, Shuming Ma, Hanwen Xu, Naoto Usuyama, Jiayu Ding, Hoifung Poon, and Furu Wei. “When an image is worth 1,024 x 1,024 words: A case study in computational pathology.” *arXiv preprint arXiv:2312.03558*, 2023.
- [WOB18] Carly B Warner, Andreina A Ottman, and Jamie N Brown. “The Role of Atomoxetine for Parkinson Disease–Related Executive Dysfunction: A Systematic Review.” *Journal of clinical psychopharmacology*, **38**(6):627–631, 2018.
- [XLZ18] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. “Unified perceptual parsing for scene understanding.” In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.
- [XRK19] Da Xu, Chuanwei Ruan, Sushant Kumar, Evren Korpeoglu, and Kannan Achan. “Self-attention with functional time representation learning.” *arXiv preprint arXiv:1911.12864*, 2019.
- [XRK20] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. “Inductive representation learning on temporal graphs.” *arXiv preprint arXiv:2002.07962*, 2020.
- [XYN22] Feng Xie, Han Yuan, Yilin Ning, Marcus Eng Hock Ong, Mengling Feng, Wynne Hsu, Bibhas Chakraborty, and Nan Liu. “Deep learning for temporal data representation in electronic health records: A systematic review of challenges and methodologies.” *Journal of biomedical informatics*, **126**:103980, 2022.
- [XZC22] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. “Simmim: A simple framework for masked image modeling.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9653–9663, 2022.
- [YAN15] Zheng Ye, Ellemarije Altena, Cristina Nombela, Charlotte R Housden, Helen Maxwell, Timothy Rittman, Chelan Huddleston, Charlotte L Rae, Ralf Regenthal, Barbara J Sahakian, et al. “Improving response inhibition in Parkinson’s disease with atomoxetine.” *Biological psychiatry*, **77**(8):740–748, 2015.

- [YLK21] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. “Vector-quantized image modeling with improved vqgan.” *arXiv preprint arXiv:2110.04627*, 2021.
- [YLW14] Hao Ye, Qi Liu, and Jia Wei. “Construction of drug network based on side effects and its application for drug repositioning.” *PloS one*, **9**(2):e87864, 2014.
- [YLZ23] Nan Ying, Yanli Lei, Tianyi Zhang, Shangqing Lyu, Chunhui Li, Sicheng Chen, Zeyu Liu, Yu Zhao, and Guanglei Zhang. “CPIA Dataset: A Comprehensive Pathological Image Analysis Dataset for Self-supervised Learning Pre-training.” *arXiv preprint arXiv:2310.17902*, 2023.
- [YYM18] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. “Hierarchical graph representation learning with differentiable pooling.” *Advances in neural information processing systems*, **31**, 2018.
- [YYW18] Jaswanth K Yella, Suryanarayana Yaddanapudi, Yunguan Wang, and Anil G Jegga. “Changing trends in computational drug repositioning.” *Pharmaceuticals*, **11**(2):57, 2018.
- [ZAL18] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. “Modeling polypharmacy side effects with graph convolutional networks.” *Bioinformatics*, **34**(13):i457–i466, 2018.
- [ZAO13] Ping Zhang, Pankaj Agarwal, and Zoran Obradovic. “Computational drug repositioning by ranking and integrating multiple data sources.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 579–594. Springer, 2013.
- [ZCN18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. “An end-to-end deep learning architecture for graph classification.” In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [ZEV15] Theresa A Zesiewicz, Marian Evatt, Camille P Vaughan, Israt Jahan, Carlos Singer, Raul Ordorica, Jason L Salemi, Jessica D Shaw, Kelly L Sullivan, et al. “Randomized, controlled pilot trial of solifenacin succinate for overactive bladder in Parkinson’s disease.” *Parkinsonism & related disorders*, **21**(5):514–520, 2015.
- [ZGA19] Yanning Zhou, Simon Graham, Navid Alemi Koohbanani, Muhammad Shaban, Pheng-Ann Heng, and Nasir Rajpoot. “Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

- [ZGG22] Yi Zheng, Rushin H Gindra, Emily J Green, Eric J Burks, Margrit Betke, Jennifer E Beane, and Vijaya B Kolachalama. “A graph-transformer for whole slide image classification.” *IEEE transactions on medical imaging*, **41**(11):3003–3015, 2022.
- [ZHW19] Difan Zou, Ziniu Hu, Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. “Layer-dependent importance sampling for training deep and large graph convolutional networks.” *Advances in neural information processing systems*, **32**, 2019.
- [ZJN23] Qingyuan Zheng, Zhengyu Jiang, Xinmiao Ni, Song Yang, Panpan Jiao, Jiejun Wu, Lin Xiong, Jingping Yuan, Jingsong Wang, Jun Jian, et al. “Machine Learning Quantified Tumor-Stroma Ratio Is an Independent Prognosticator in Muscle-Invasive Bladder Cancer.” *International Journal of Molecular Sciences*, **24**(3):2746, 2023.
- [ZMP16] Shunhua Zhang, Cong Ma, Haijie Pang, Fanpeng Zeng, Long Cheng, Binbin Fang, Jia Ma, Ying Shi, Haiyu Hong, Jianyan Chen, et al. “Arsenic trioxide suppresses cell growth and migration via inhibition of miR-27a in breast cancer cells.” *Biochemical and biophysical research communications*, **469**(1):55–61, 2016.
- [ZMW19] Si Zheng, Hetong Ma, Jiayang Wang, and Jiao Li. “A computational bipartite graph-based drug repurposing method.” In *Computational Methods for Drug Repurposing*, pp. 115–127. Springer, 2019.
- [ZMZ22] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. “Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18802–18812, 2022.
- [ZWW21] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. “ibot: Image bert pre-training with online tokenizer.” *arXiv preprint arXiv:2111.07832*, 2021.
- [ZXH15] Jinghe Zhang, Haoyi Xiong, Yu Huang, Hao Wu, Kevin Leach, and Laura E Barnes. “M-SEQ: Early detection of anxiety and depression via temporal orders of diagnoses in electronic health data.” In *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2569–2577. IEEE, 2015.
- [ZYF20] Yu Zhao, Fan Yang, Yuqi Fang, Hailing Liu, Niyun Zhou, Jun Zhang, Jiarui Sun, Sen Yang, Bjoern Menze, Xinjuan Fan, et al. “Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4837–4846, 2020.

- [ZZL19] Xiangxiang Zeng, Siyi Zhu, Xiangrong Liu, Yadi Zhou, Ruth Nussinov, and Feixiong Cheng. “deepDR: a network-based deep learning approach to in silico drug repositioning.” *Bioinformatics*, 2019.