UNIVERSITY OF BIRMINGHAM

DOCTORAL THESIS

Graph Representation Learning: Algorithm Development and its Application to Early Drug Discovery

---

Author:

Yang YUE

Supervisors:

Dr. Shan HE, Dr. Leandro Lei MINKU, and Dr. Alan JONES



A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Department of Computer Science

January 2025

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

UNIVERSITY OF BIRMINGHAM

Abstract

School of Computer Science

Department of Computer Science

Doctor of Philosophy

Graph Representation Learning: Algorithm Development and its Application to Early Drug Discovery

by Yang YUE

Drug discovery is a time- and cost-consuming process because of the complex biological mechanisms of the diseases and interactions between the drug and biological system. In the early (candidate selection) stage of drug discovery, machine learning, especially the graph representation learning (GRL)-based approaches have drawn wide-range attention. This is due to their powerful capability to flexibly handle various molecular and biological interaction information at different data scales, enabling effective reduction of the search space of following wet experiment validations based on efficient candidate screenings. Focusing on one of the most popular GRL approaches graph neural networks (GNNs) and relevant techniques, this thesis provides a comprehensive examination of a series of specialized designs on multiple crucial early drug discovery tasks, with the following contributions:

- Based on the large-scale heterogeneous drug-disease-target (DDT) relationship graphs, the examination of feasibility of non-Euclidean hyperbolic representation learning for identifying candidate targets for given drug and disease types.

- Based on the large-scale drug-target interaction (DTI) graphs, the examination of feasibility of meta-path-based representation learning for drug-drug therapeutic synergy effect predictions with the help of adverse effect (AE) information.

- Based on the small-scale individual protein molecular graphs, the examination of feasibility of a pre-trained geometric equivariant framework for predicting the protein binding affinity change caused by amino acid (AA) mutations with the mixture of atom- and residue-scale molecular representation information.

- Based on the small-scale protein molecular graphs, the examination of feasibility of a more general and efficient geometric framework based on coarse-grained (CG) molecular representation learning for broader candidate protein screening tasks.

In summary, this thesis contributes towards a greater understanding of GRL applications on early drug discovery, ranging from learning macro-biological interaction graphs to micro-molecular structure graphs, to enable further potential biological mechanism exploration and experimental validations.

Acknowledgements

I would like to take this opportunity to thank my primary supervisor Dr. Shan He. His insightful suggestions and expertise lead me into the path of pursuing the PhD and accompany throughout the whole course. He gives me a flexible and relaxed academic environment where I can explore whatever interests me within my PhD topic. Without these, I would not achieve the outcomes filled in this thesis. I would also like to thank him for all of his help during the past three years.

I would like to thank my secondary supervisors Dr. Leandro Lei Minku and Dr. Alan Jones and thesis group members Dr. Jinming Duan and Dr. Rami Bahsoon, for their constructive comments and continuous support on my thesis and progress.

I wish to thank Prof. Hujun Yin from The University of Manchester and Dr. Jian Liu from The University of Birmingham for reading my thesis and being the examiners of my PhD viva. I thank them for their kind dedication. Their advice and feedback are highly useful and important to further improve the quality and completeness of this thesis. I also wish to thank Dr. Shuo Wang for hosting my PhD viva process.

Next, I wish to extend my thanks to Dr. Shu Li, as a computational biologist, she teaches me a lot of biological knowledge, to let me know how to better formulate a life science machine learning task with the support of reasonable biological hypothesis. I also would like to thank Dr. Yihua Cheng and Dr. David McDonald, for their provided crucial collaboration and guidance in my academic career. Through discussing and working with them, I have learnt how to better summarize the scientific problems and conduct research in a more organized manner.

I would now like to thank all of my friends and colleagues around. They give me various support and help when I feel lowest, helping me to go through the tough time in my research.

Finally, I would like to express my gratitude to my family and my girl friend, they provide their warm support and encouragement over the whole PhD pursuit period, making me always move forward.

故余虽愚，卒获有所闻。

Contents

Abstract **i**

Acknowledgements **iii**

1 Introduction **1**

1.1 Graph representation learning for early drug discovery at different data scales 1

1.2 Research Questions 2

1.2.1 Hyperbolic Embedding of Heterogeneous Drug-Disease-Target Networks/Graphs for Target Identification 3

1.2.2 Meta-path-based Embedding of Heterogeneous Drug-Target Networks/Graphs for Drug-Drug Synergistic Therapeutic Effect Prediction 4

1.2.3 Geometric Equivariant Embedding of Protein-Protein Complex Graphs for Screening Candidate Proteins with Stronger Binding Intensities to the Goal Target 5

1.2.4 Coarse-grained-scale Protein Complex Geometric Representation for More General Candidate Screening or Property Predictions 6

1.3 Thesis Contributions 7

1.4 List of Publications 9

1.5 Thesis Organisation 10

2 Literature Review **12**

2.1 Graph Representation Learning for Biological Interaction Network-based Target Identification 12

2.1.1 Basic Definitions of Graph and Graph Representation Learning 13

2.1.2 Definitions of Biological Interaction Network-based Target Identification . . . 14
2.1.3 Some Representative Task Formulations and Solutions in Target Identification . . . 15
2.1.4 Hyperbolic Space Embedding . . . 18
2.2 Graph Neural Network-based Heterogeneous Network Embedding for Drug-Drug Therapeutic Effect Prediction . . . 19
2.2.1 Definitions of Heterogeneous Interaction Graph-based Drug-Drug Therapeutic Effect Prediction . . . 20
2.2.2 Meta-path-based Graph Neural Network for Heterogeneous Biological Graph Learning . . . 20
2.3 Geometric Equivariant Representation Learning for Prediction of Binding Affinity Change Cause by Amino Acid Mutations . . . 22
2.3.1 Definitions of Graph Neural Network-based Geometric Equivariant Representation Learning . . . 22
2.4 Coarse-grained Protein Complex Geometric Representation Learning for General Candidate Screening . . . 25
2.4.1 Definitions of Geometric Graph Representation Learning at The Coarse-grained Particle Scale . . . 25
2.5 Chapter Summary . . . 27

3 FLONE: Fully Lorentz network embedding for identification of novel drug targets 28
3.1 Introduction . . . 28
3.2 Materials and methods . . . 32
3.2.1 Datasets . . . 32
3.2.2 The basic definition of the Lorentz model . . . 33
3.2.3 Description of the FLONE method . . . 34
3.2.4 Description of FuLLiT . . . 37
3.2.5 The hyperbolic encoders for fusing drug and target similarity information . . . 39
3.2.6 Model training and optimization . . . 41

3.3 Results and discussion 41
3.3.1 Hyperbolicity of the DDT networks and scalability analysis . . 41
3.3.2 Model evaluation settings 42
3.3.3 Experiments without external domain knowledge injection . . 44
3.3.4 Fusing domain knowledge for FLONE 47
3.3.5 Extra ablation study 48
3.3.6 Visualization of the embedding spatial layout 51
3.4 Chapter Summary 52

4 Muthene: Improving therapeutic synergy score predictions with adverse effects using multi-task heterogeneous network embedding 53
4.1 Introduction 53
4.2 Materials 56
4.2.1 Datasets 56
4.3 Methods 57
4.3.1 Construction of the heterogeneous therapeutic effect network . 57
4.3.2 Overview of Muthene 59
4.3.3 TE-related meta-path learning module 59
4.3.4 Integration of drug chemical information 64
4.3.5 The adverse effect probability score prediction module 65
4.3.6 The therapeutic synergy score prediction module 65
4.3.7 Model evaluation settings 67
4.4 Results 68
4.4.1 AE prediction task benefits synergy score predictions 68
4.4.2 The effectiveness of selected network components 69
4.4.3 The effects of the weights between the two losses 72
4.4.4 MoAs Elucidation: A Case Study 73
4.5 Chapter Summary 75

5 MpbPPI: A geometric equivariant approach for measuring mutation effects on protein interactions 76
5.1 Introduction 77
5.2 Results 81

5.2.1 Overview of MpbPPI 81
5.2.2 MpbPPI yields accurate PPI $\Delta\Delta G$ prediction 83
5.3 Methods 91
5.3.1 Generation of the refined residue-level contact graphs for protein-protein complexes 91
5.3.2 Graph edge features 93
5.3.3 The GEE encoder and its pre-training techniques 93
5.3.4 The prediction of downstream PPI $\Delta\Delta G$ 96
5.4 Chapter Summary 97

6 MCGLPPI: Integration of molecular coarse-grained model into geometric representation learning for protein-protein complex property prediction 99
6.1 Introduction 100
6.2 Results 103
6.2.1 Overview of the proposed CG-scale complex geometric learning framework 103
6.2.2 Force field parameter and CG-scale complex graph generation . 105
6.2.3 CG-scale geometric representation learning 107
6.2.4 DDI-based CG-scale graph encoder pre-training 108
6.2.5 MCGLPPI saves computational cost while keeping competitive performance 108
6.2.6 The binding affinity prediction of formation of strict dimers . . 109
6.2.7 The effectiveness of MCGLPPI on more complex PPI patterns . 111
6.2.8 The prediction results for protein-protein interface classification 113
6.2.9 The investigation of CG-scale pre-training techniques on different tasks 115
6.2.10 Performance of the geometries considered in CG-scale complex graphs 118
6.2.11 Influence of graph cropping on overall model efficiency 119
6.3 Methods 120
6.3.1 The detailed curation process for the 3DID pre-training dataset 120
6.3.2 MARTINI-based geometric parameter generation 120

6.3.3 The construction of CG-scale protein complex graph and its cropping function . . . 122
6.3.4 The CG-scale representation learning for complex overall property prediction . . . 125
6.3.5 The DDI-based CG graph encoder pre-training technique . . . 126
6.4 Chapter Summary . . . 126

7 Discussion **128**
7.1 Contributions & Conclusions . . . 128
7.1.1 Ranking of Novel Drug Targets with the Consideration of Disease Types Based on Hyperbolic Lorentz Space Representation Learning . . . 128
7.1.2 Screening of Drug Pairs with Better Therapeutic Effects Based on Meta-path-based Heterogeneous Network Representation Learning and Auxiliary Adverse Effect Learning . . . 129
7.1.3 Identification of Candidate Protein Complex Mutants with Stronger Binding Affinities Based on Pretraining-based Equivairant Geometric Representation Learning . . . 130
7.1.4 Screening of Protein (Complex) Candidates towards Broader Property Criteria Based on Molecular Coarse-Grained Force Field Modelling . . . 133
7.2 Future Directions . . . 134
7.2.1 Incorporate Non-Euclidean Space Embedding with Positive Curvatures for Learning Network Sub-structures . . . 134
7.2.2 Incorporate More Biological Plausible Meta-path Varieties for MoA Modelling . . . 135
7.2.3 Incorporate Biochemical and Evolutionary Related Information into Geometric Representation Learning for Mutant Screening . . . 135
7.2.4 Support More Molecular Coarse-Grained Force Fields and Protein-Protein Complex Generation Tools for Broader Protein Complex Screening . . . 135

7.2.5 Incorporate Inductive Target Protein Initial Features to Increase the Generalization Capability of Models 136

A Supplementary for Chapter 3 **138**

A.1 The triplet screening based on Jaccard coefficient 138

A.2 The GO-PPI network and its encoder 138

A.3 Statistics of the independent test sets 139

A.4 Detailed experimental setup of NeoDTI and KGE_RF 140

A.5 Fusing domain knowledge for the Euclidean-based model 140

A.6 The advantages of the triplet-wise association prediction 141

A.7 The projection details of Lorentz and Euclidean embeddings 143

B Supplementary for Chapter 4 **144**

B.1 The illustration of steps of collecting our dataset 144

B.2 Threshold selection for drug-drug TE relationships 144

B.3 Model running environment and hyper-parameter settings 145

B.4 Discussion of involved three representative methods 146

B.5 Detailed description of Mut-GIN 147

B.6 Ablation study for dimension of meta-path embeddings 148

C Supplementary for Chapter 5 **150**

C.1 Pre-training dataset collection procedure 150

C.2 The generation of the mutant protein-protein complex structure 151

C.3 Baseline Methods 152

C.4 The RMSE and MAE results of MpbPPI with different mutant structure sources under the mutation-level tenfold cross-validation 154

C.5 Further result providing and analysis of MpbPPI with different mutant structure sources 154

C.6 Experiment about correlating sequence representation difference to the PPI $\Delta\Delta G$ value 156

C.7 Experiment about reproducing results of some representative baseline methods based on available information 157

C.8 The RMSE and MAE results of MpbPPI with different pre-training task combinations under the mutation-level tenfold cross-validation . 158
C.9 Detailed description of the protein-protein complex type-based data splitting 159
C.10 The illustrative example of the predicted outcomes of MpbPPI 159
C.11 The RMSE and MAE results of MpbPPI with different pre-training task combinations under the wild-type protein-protein complex type-based cross-validation 160
C.12 The discussion of some optional features for $\Delta\Delta G$ predictions 160
C.13 Detailed description of the residue node feature calculation 164
C.14 Equations of the GVP-GNN backbone 165
C.15 Detailed description of the devised pre-training strategies 165
C.16 Details of the interface identification based on PyMOL 167
C.17 Summary of implementation tools for MpbPPI 168

D Supplementary for Chapter 6 **169**

D.1 The version characteristics of MARTINI22 and MARTINI3 169
D.2 The statistics of graph nodes and edges of the MANY/DC dataset . . . 170
D.3 Further extension to protein complex $\Delta\Delta G$ predictions based on simple modifications 171
D.4 Detailed curation process of the PDBbind-strict-dimer dataset 174
D.5 The default hyper-parameters of involved approaches in comparison . 174
D.6 Investigation about the influence of hidden feature dimensions on overall performance of MCGLPPI 176
D.7 The experimental results on more challenging experimental settings . 176
D.8 Model stability test using more accessible AlphaFold-generated structures 178
D.9 The other experimental results for the evaluation of the DDI-based pre-training technique 179
D.10 More specific performance difference analysis across different MARTINI versions 179

D.11 The potential extension of using corresponding force field parameters in other scales 181
D.12 The definition of generation of a one-hot representation 182
D.13 Detailed description of allocation of sparse angular features to specific bead nodes 182
D.14 The equations of the GearNet-Edge protein encoder 183
D.15 The details about the CG-scale pre-training technique 184
D.16 Summary of implementation tools for MCGLPPI 186

Bibliography **188**

List of Figures

2.1 The geometric shape of the Lorentz feature space 18

3.1 Illustration of the hierarchies in the DDT network 31
3.2 Illustration of FLONE 36
3.3 Visualization of the 2D embedding spatial layout of 128-dimension FLONE$_{ECFP-SEQ}$ and its Euclidean counterpart 51

4.1 Illustration of Muthene 60
4.2 Different model evaluation results given various weight ratio R 72
4.3 The network constituted by all DTTTD instances shared by *Vincristine* and *Dasatinib* 74

5.1 The flowchart of the MpbPPI framework 82
5.2 The illustration of data flow within the MpbPPI framework 84
5.3 MpbPPI outperformed other involved methods for PPI $\Delta\Delta G$ prediction under the five-time WT protein-protein complex-based cross-validations 90

6.1 The flowchart of the MCGLPPI framework 104
6.2 MARTINI-based CG-scale representation of protein structure 106
6.3 The investigation of performance influence contributed by the DDI-based pre-training, crucial graph geometric characterization components, and cropping function of MCGLPPI 117

7.1 Comparison of mutant PPI structures from various mutant generation tools 132

B.1 The illustration of collecting relevant data for a drug-drug-cell line pair 145

B.2 Different model evaluation results given various meta-path embedding dimensions 148

C.1 The R_P results of involved MpbPPI variants under the mutation-level tenfold cross-validation on the four PPI $\Delta\Delta G$ datasets 155

C.2 The test set prediction results of MpbPPI-FoldX under the mutation-level tenfold cross-validation on M1101 (unit: kcal·mol-1) 156

List of Tables

3.1 Comparison results of involved methods under the 16 embedding dimension . . . 45
3.2 Comparison results of involved methods under the 128 embedding dimension . . . 46
3.3 Comparison results of involved methods under 16 and 128 embedding dimensions based on the DTINet dataset . . . 49
3.4 Comparison results of involved methods in the ablation study based on the DTINet dataset . . . 50

4.1 The summary of our collected dataset . . . 57
4.2 Description of biological meanings of four pre-defined meta-paths . . . 58
4.3 Description of the involved comparison methods . . . 69
4.4 The averaging evaluation results of involved methods over the five independent repeats . . . 70
4.5 Ablation study results based on involved Muthene variants . . . 71

5.1 Comparison of the R_P of the baseline methods on downstream PPI $\Delta\Delta G$ datasets . . . 85
5.2 Comparison of the R_P of five MpbPPI-FoldX variants on each downstream dataset . . . 88

6.1 Test performance and computational cost of different approaches at different scales on the 915-subset of the PDBbind-strict-dimer dataset based on one A100 GPU 40GB . . . 110
6.2 Test performance and computational cost of different approaches at different scales on the 451-subset of the curated ATLAS dataset based on one A100 GPU 40GB . . . 112

6.3 Test performance and computational cost of different approaches at different scales on the 5535-151-subset of the MANY/DC dataset based on one A100 GPU 40GB 114

A.1 The statistics of each independent repeat/test set in the comparison experiments 140
A.2 The statistics of sample numbers of the different testing parts in extra ablation study 140
A.3 Comparison results of the involved Euclidean models under 16 and 128 embedding dimensions based on the DTINet dataset 141
A.4 The evaluation results on the test drug-disease-target triplets corresponding to positive diseases and negative diseases (on DTINet) . . . 143

B.1 The hyper-parameters shared by all Muthene variants in the comparison experiment 146
B.2 The hyper-parameters shared by all DeepSynergy variants in the comparison experiment 146

C.1 The RMSE and MAE results of MpbPPI with different mutant structure sources under the mutation-level tenfold cross-validation on the four PPI $\Delta\Delta G$ datasets 154
C.2 R_P results of the re-produced representative methods 157
C.3 The RMSE and MAE results of involved MpbPPI variants under the mutation-level tenfold cross-validation 158
C.4 The RMSE and MAE results of involved methods under the cross-validation that splits data based on the wild-type protein-protein complex types 161

D.1 The statistics of nodes and edges in each set of the MANY/DC dataset 170
D.2 the $\Delta\Delta G$ prediction results based on the AB-bind dataset 173
D.3 The experimental results based on structural homology reduction settings 177
D.4 The experimental results based on AlphaFold-generated structures . . 179
D.5 The other experimental results of the pre-training at different scales . . 180

List of Abbreviations

3D	three-**D**imensional
3DID	three-**D**imensional **I**nteracting **D**omains
AA	**A**mino **A**cid
AE	**A**dverse **E**ffect
AUROC	the **A**rea **U**nder the **R**eceiver **O**perating characteristic **C**urve
AUPR	the **A**rea **U**nder the **P**recision-**R**ecall curve
BCE	**B**inary **C**ross **E**ntropy
BiGRU	**Bi**-directional **G**ated **R**ecurrent **U**nit
CG	**C**oarse-**G**rained
CV	**C**ross-**V**alidation
DTI	**D**rug **T**arget **I**nteraction
DDT	**D**rug-**D**iease-**T**arget
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
DDI	**D**omain **D**omain **I**nteraction
ECFP	**E**xtended **C**onnectivity **F**inger**P**rint
FF	**F**orce **F**ield
GRL	**G**raph **R**epresentation **L**earning
GNN	**G**raph **N**eural **N**etwork
GAT	**G**raph **AT**tention Network
GIN	**G**raph **I**somorphism **N**etwork
GEE	**G**eometric **E**quivariant **E**ncoder
GBT	**G**radient-**B**oosting **T**rees
GPCR	**G** **P**rotein-**C**oupled **R**eceptor
GPU	**G**raphics **P**rocessing **U**nit

HSA	**H**ighest **S**ingle **A**gent
KG	**K**nowledge **G**raph
KGE	**K**nowledge **G**raph **E**mbedding
KGC	**K**nowledge **G**raph **C**ompletion
KNN	**K**-**N**earest **N**eighbor
LRRK	**L**eucine-**R**ich **R**epeat **K**inase
MRR	**M**ean **R**eciprocal **R**ank
MLP	**M**ulti-**L**ayer **P**erception
MT	**M**utant-**T**ype
MD	**M**olecular **D**ynamics
ML	**M**achine **L**earning
MoA	**M**echanisms **o**f **A**ctions
MSE	**M**ean **S**quare **E**rror
MAE	**M**ean **A**bsolute **E**rror
PPI	**P**rotein **P**rotein **I**nteraction
PDB	**P**rotein **D**ata **B**ank
Pearson	**Pearson**'s correlation coefficient
pMHC	an antigenic **p**eptide presented by the **M**ajor **H**istocompatibility **C**omplex
PLM	**P**rotein **L**anguage **M**odel
ReLU	**Re**ctified **L**inear **U**nit
RIPK	**R**eceptor-**I**nteracting serine/threonine-**P**rotein **K**inase
SASA	**S**olvent-**A**cessible **S**urface **A**rea
TE	**T**herapeutic **E**ffect
TCR	**T** **C**ell **R**eceptor
WT	**W**ild-**T**ype
ZIP	**Z**ero **I**nteraction **P**otency

Dedicated to everyone who loves and supports me

Chapter 1

Introduction

1.1 Graph representation learning for early drug discovery at different data scales

Discovering and developing a novel drug from scratch is a time-consuming process which could take 12-15 years with over 1 billion dollars (Eslami Manoochehri and Nourani, 2020). With the help of rapid advance of computational techniques, the cost of early screenings that aim to identify potentially effective drug and target candidates for further wet experimental functional validations (i.e., early drug discovery) can be greatly reduced. Because these *in-silico* methods can provide more efficient calculation and analysis based on available data, to effectively narrow down corresponding candidate search space (Luo et al., 2017).

In recent years, the machine learning-based representation learning approaches especially those based on graph modelling, have drawn wide-spread attention. This is because many crucial data structures used in early drug discovery, ranging from large-scale biological interaction networks [e,g, drug-target interaction (DTI) (Luo et al., 2017) and protein-protein interaction (PPI) networks (Szklarczyk et al., 2015)] to small-scale individual drug molecular (Wu et al., 2021) or antibody-antigen protein complex structures (Kong, Huang, and Liu, 2022) can be naturally modelled as a graph. For example, each protein target entity, heavy atom, and amino acid (AA) residue can be treated as a graph node in PPI networks, small-molecular drugs, and macro-molecular complexes, respectively. On top of this, through various prior-based pre-defined regulations, the edges between nodes can be established to capture corresponding known PPI relationships, chemical bonds, as well as geometric

spatial relationships. Further, the graph representation learning (GRL) is favoured as it can effectively transform or encode these non-Euclidean graph data into a set of easy-to-execute low-dimensional representations (i.e, embeddings) for the subsequent processing (Kipf and Welling, 2016).

To encode the aforementioned graph data such as DTI and PPI networks, graph neural networks (GNNs) are one of the most promising choices (Liu et al., 2021; Cai et al., 2024). Compared to some traditional GRL methods including ones based on heuristic algorithm (Jia et al., 2012), matrix factorization (Belkin and Niyogi, 2001), and random walk (Perozzi, Al-Rfou, and Skiena, 2014), GNNs can offer more powerful adaption on the efficiency, inductive learning ability, as well as capability to handle more complex graph structures that could consist of various node and edge types and features (Zhang, 2022).

As a general GRL architecture, GNN is already demonstrated its stability in several other application scenarios where the source data can be typically constructed as a graph [e.g., search and advertising recommendation systems (Sun et al., 2011; He et al., 2017)]. However, how to properly and effectively design the graph structures and corresponding GNN-based frameworks under aforementioned (screening) tasks in early drug discovery supported with reasonable biological understandings still needs further exploration.

1.2 Research Questions

Due to the complexity of the biological systems and the hierarchy of source data ranged from large-scale to small-scale, how to utilize concise yet accurate GNN-related GRL characterization, to effectively solve a series of important drug and target candidate screening tasks in early drug discovery based on available data at different scales, are the overarching research question of this thesis.

According to the scale of required source data (from large-scale biological interaction networks to small-scale molecular structure data), the next subsections introduce four detailed research questions that this thesis intends to answer in detail.

1.2.1 Hyperbolic Embedding of Heterogeneous Drug-Disease-Target Networks/Graphs for Target Identification

A target is a biological molecule, such as a protein or gene, that is involved in a disease process and can be modulated by a drug or therapeutic intervention (Lindsay, 2003). Target identification (e.g., protein target identification) is an important task in early drug discovery. The aim of target identification is to explore the known relationships between drug, target, and disease (entities) organized in heterogeneous biological networks (i.e., graphs where nodes represent these entities and edges give known inter-entity relationships), for predicting potential novel drug targets (Moon et al., 2021). However, most of existing GRL methods cannot explicitly consider disease types when predicting targets. More importantly, drug-disease-target (DDT) networks could exhibit complex hierarchical sub-structures which provide extra topological information about the interaction relationships in the DDT network, but these methods, especially those based on Euclidean space embedding could not fully capture such topological information, which might lead to sub-optimal prediction results.

Based on this, the first research question of this thesis is:

- Within the large-scale DDT networks that could contain hierarchies, whether hyperbolic space, which theoretically has more capacity to capture hierarchical topology structures (Ganea, Bécigneul, and Hofmann, 2018), can better predict potential targets given the drug and disease type? If it is the case, how to introduce the more prior knowledge into these entities (e.g., similarity features for corresponding drug and target entities) within the hyperbolic space to further improve the prediction accuracy on previously unknown entities during model training?

By answer this question, we hope to develop a proof-of-concept approach that provides finer and more accurate control on target predictions based on biological interaction networks in early drug discovery.

A flexible framework for low-dimensional hyperbolic representation learning for (protein) target identification based on explicit consideration of disease types within DDT networks, is introduced in chapter 3.

The details of the approach are described in section 3.2.

1.2.2 Meta-path-based Embedding of Heterogeneous Drug-Target Networks/Graphs for Drug-Drug Synergistic Therapeutic Effect Prediction

In addition to the target identification, another important task in early drug discovery is to perform fast drug screening within the entire drug library or more complex assay system to narrow down the scope of candidate drugs. Within this task, a critical problem is to predict synergistic therapeutic effects (TEs) brought by drug combination treatments, for further screening drug combinations which could rise better treatment effects in some cancer cell lines.

Specifically, the drug combination treatments could trigger pharmacological TEs (Preuer et al., 2018) and adverse effects (AEs) (Ryu, Kim, and Lee, 2018). Many computational methods have been developed to predict TEs or AEs from drug-drug interactions. Based on these predictions, drug combinations with better TEs or less AEs could be preliminarily selected. However, most of the methods treat the AEs and TEs predictions as two separated tasks, ignoring the potential mechanistic commonalities shared between them.

This gives rise to the second research question:

- Based on the large-scale drug-target interaction (DTI) networks, whether we can mimic the biological interaction commonalities of drug pairs for both TEs and AEs using the pre-defined interaction path patterns which use (pre-defined) sequences of node and edge types that capture meaningful relationships in heterogeneous graphs [i.e., meta-paths (Fu et al., 2020)], and use multi-task learning (Zhang and Yang, 2021) to boost the TE predictions based on learning the captured AE information from these commonalities?

By answer this question, we hope to develop a proof-of-concept approach that provides more accurate synergistic therapeutic drug combination screenings based on biological interaction networks and adverse effect information in early drug discovery.

Chapter 4 proposes a solution to this question, that is described in section 4.3 and validated in section 4.4.

1.2.3 Geometric Equivariant Embedding of Protein-Protein Complex Graphs for Screening Candidate Proteins with Stronger Binding Intensities to the Goal Target

In the context of screening of computationally-designed peptide drugs or antibodies binding to a protein target (this will result in a protein complex further explained in section 2.3.1), large-scale interaction network-based GRL methods cannot provide enough fine information. In this context the models need to more carefully distinguish the design with stronger functional binding intensity (i.e., binding affinity) to the target, from a large amount of candidate binders with very similar amino acid (AA) sequences or three-dimensional (3D) geometric structures (i.e., conformations) [probably with only a few AA type substitutions or mutations, i.e., there could be only several AA type differences in the entire protein primary structure (Liu et al., 2021)].

Protein structures determine their functions (Petsko and Ringe, 2004). Based on this biological principle, geometric GRL methods, working on small-scale individual molecular data, which model protein or protein complex structures as a graph and capture the subtle structural change among similar designs, become an intuitive solution for solving aforementioned screenings. However, many relevant methods face the difficulty of sufficient representation of complex protein structures.

Specifically, proteins are macro-molecules containing complex hierarchical structures. In other words, each protein contains the AA sequence chain that can fold into corresponding 3D conformation in solution, and further each AA includes smaller-scale Carbon-Hydrogen-Oxygen constituted backbone atoms and side-chain atoms determining the chemical properties of current AA residue (Petsko and Ringe, 2004). Such structure nature brings challenges on expressive learning for such methods.

Based on this, how to capture the complex inter-atom and inter-residue relationships within protein complexes, which are formed from designed proteins binding to the target, to achieve the aforementioned screening becomes an unavoidable problem. Additionally, due to the expensive cost on acquiring experimentally validated

labelled data for (supervised) model training, another challenge is how to give a better generalization capability for GRL methods under limited training samples.

In light of this, the third research question is:

- Whether the specially-designed pre-training aiming to learn the general geometric regulations on available protein structures (not involving labels) (Liu et al., 2021) combined with finer inter-atom and inter-residue relationship characterization can bring better screening ability into geometric GRL methods for above task?

By answer this question, we hope to develop a proof-of-concept approach that provides more accurate similar candidate binder screenings based on protein structures in early drug discovery.

A robust framework for screening mutations sharing higher binding affinities to the protein target (compared to other candidate binders) and the corresponding prediction results are provided in section 5.2 and section 5.3, respectively.

1.2.4 Coarse-grained-scale Protein Complex Geometric Representation for More General Candidate Screening or Property Predictions

After gaining a deeper insight of using GRL techniques to characterize the geometric details of the protein (complex) for downstream drug discovery-related screening or property predictions in section 1.2.3, a further problem rises:

In order to model the protein (complex), GRL methods usually need to establish a protein graph at the atom-scale (in which each heavy atom is treated as a graph node) (Liu et al., 2021) or residue-scale (in which each AA is a node) (Jiang et al., 2023), for preserving the fundamental geometric and interaction information to be encoded by corresponding GRL methods. However, each representation has its own trade-offs. Atom-scale-based representation could be more detailed while demanding expensive computational resources to manage thousands of nodes, limiting their application to large protein-protein interaction (PPI) systems. Residue-scale-based representation could be more computationally tractable (as positioned at a higher particle hierarchy compared to ones at the atom-scale) but may overlook critical binding details that influence binding specificity and affinity. On this basis,

we hypothesize that mesoscopic representation functioning between the atom-scale and residue-scale could strike a better trade-off between computational overhead and finely-described inter-particle interaction details, which can facilitate broader application in protein complex screening or property predictions.

Naturally, the fourth research question is:

- Whether we can realize a versatile yet efficient geometric GRL framework that works on concise and chemical-plausible mesoscopic-scale between the atom-scale and residue-scale, on which the aforementioned objective can be achieved?

By answer this question, we hope to develop a proof-of-concept approach that provides support for multiple protein candidate screening tasks based on mesoscopic-represented protein structures in early drug discovery.

Aimed by this, a GRL framework based on mesoscopic molecular coarse-grained (CG)-scale geometric modeling is implemented and widely examined in chapter 6.

1.3 Thesis Contributions

The main purpose of the works proposed in this thesis are to provide frameworks, which are designed to give several novel insights on solving multiple crucial tasks corresponding to source data with different scales (from large biological networks to small molecular structures) in early drug discovery. These frameworks are able to generate specified predictions from the modern GRL techniques based on corresponding insights, which can be referred and analyzed by domain experts to provide supportive suggestions for further drug or target candidate screenings (in early drug discovery). Specifically, the works contribute to the field in the following four aspects:

- We propose a novel GRL approach fully working at a Lorentz space, a variety of the hyperbolic space (Chen et al., 2021). This approach achieves better target identification capability when specifying drug and disease types compared to the GRL counterparts running at the Euclidean space (chapter 3). This work was published in *Bioinformatics Advances* (Yue et al., 2023a).

- We develop a novel multi-task heterogeneous DTI network embedder model that performs TE and AE learning tasks simultaneously based on learning DTI commonalities from pre-defined meta-paths. By introducing the AE prediction as an auxiliary task, the model generates more accurate TE predictions than the standard single-task learning (i.e., TE prediction) methods (chapter 4). The method was published in *Briefings in Bioinformatics* (Yue et al., 2023b).

- The task of screening candidate mutation designs possessing stronger affinities to the protein target is formulated as the prediction of binding affinity change from the wild-type binding complex to the mutant ones (i.e., measuring the binding affinity difference amount deviating from the original design template binding to the protein target). On top of this, By employing a geometric equivariant GRL technique enabling the learnt protein representation insensitive to the imposition of protein's 3D rigid motions (Jing et al., 2020), we robustly characterize the different-scale geometric interactive information between residues, between side chain atoms, and between side chain and residue backbone atoms within a residue-scale-based protein complex graph. Based on the characteristics of the devised graph, a specialized self-supervised pre-training strategy (Liu et al., 2021) is further proposed to leverage the prior geometric information of available protein structures to give accurate final predictions (chapter 5). This geometric approach was published in *Briefings in Bioinformatics* (Yue et al., 2023c).

- We implement a more general geometric GRL framework, which can handle the binding affinity predictions, the binding affinity change predictions caused by AA mutations, and the interaction interface classifications for candidate protein complexes. This framework works on a mesoscopic CG-scale theoretically supported by the MARTINI molecular CG models which are already demonstrated their accuracy on concise description of various PPI systems (Souza et al., 2021). Through the specialized design of the protein complex graph and corresponding supporting techniques (e.g., the protein graph embedder and its pre-training strategy) on the CG-scale, this framework achieves

a better trade-off between computational cost and chemically-plausible interaction detail descriptions compared to its counterpart at the atom- and residue-scale (chapter 6). This framework was published in *Nature Communications* (Yue et al., 2024).

- Based on the algorithms and methods proposed in this thesis, we have developed the following open-source software:
  - FLONE (`https://github.com/arantir123/FLONE`);
  - Muthene (`https://github.com/arantir123/Muthene`);
  - MpbPPI (`https://github.com/arantir123/MpbPPI`);
  - MCGLPPI (`https://github.com/arantir123/MCGLPPI`).

1.4 List of Publications

This thesis summarizes the contributions of the following publications:

- **Yue, Y.**, McDonald, D., Hao, L., Lei, H., Butler, M. S., & He, S. (2023). FLONE: fully Lorentz network embedding for inferring novel drug targets. *Bioinformatics Advances*, 3(1), vbad066.
- **Yue, Y.**, Liu, Y., Hao, L., Lei, H., & He, S. (2023). Improving therapeutic synergy score predictions with adverse effects using multi-task heterogeneous network learning. *Briefings in Bioinformatics*, 24(1), bbac564.
- **Yue, Y.**, Li, S., Wang, L., Liu, H., Tong, H. H., & He, S. (2023). MpbPPI: a multi-task pre-training-based equivariant approach for the prediction of the effect of amino acid mutations on protein–protein interactions. *Briefings in Bioinformatics*, 24(5), bbad310.
- **Yue, Y.**, Li, S., Cheng, Y., Wang, L., Hou, T., Zhu, Z., & He, S. (2024). Integration of molecular coarse-grained model into geometric representation learning framework for protein-protein complex property prediction. *Nature Communications*, 15(1), 9629.

- **Yue, Y.**, Cheng, Y., Marquet, C., Xiao, C., Guo, J., Li, S., & He, S. (2025). Meta-Learning Enables Complex Cluster-Specific Few-Shot Binding Affinity Prediction for Protein–Protein Interactions. *Journal of Chemical Information and Modeling*, 65(2), 580-588.

1.5 Thesis Organisation

This thesis consists of seven chapters, focusing on solving a series of important tasks (organized based on the corresponding source data scales) in early drug discovery using novel GNN-related GRL techniques. The first chapter introduces the background and the importance of this topic. I stated the research questions to be studied and summarized the contributions.

In chapter 2, I reviewed the important literature in related fields. Section 2.1 introduces the basic definition of target identification and commonly-used relevant approaches. Then section 2.2 focuses on graph embedding techniques specifically for drug-drug synergistic therapeutic effect predictions. Next section 2.3 focuses on 3D geometric equivariant GRL methods for parsing protein complex structures for predictions of binding affinity change brought by AA mutations. Section 2.4 reviews the conventional ways of protein graph constructions at the atom-scale and residue-scale and introduces the basic definition of molecular coarse-grained modelling using MARTINI engines.

In chapter 3, hyperbolic space embedding for learning the relationships between drug, disease, and target entities was studied. Section 3.1 describes the motivation and defines the problem. Section 3.2 proposes a novel hyperbolic space-based knowledge graph completion approach with optional drug and target prior feature encoders (e.g., hyperbolic GNNs) that identifies/ranks the potential targets from the available candidates conditioned on given drug and disease. The approach was validated on two curated DDT relationship network datasets detailed in section 3.3.

In chapter 4, based on a multi-task GNN-based heterogeneous graph embedding framework, the drug-drug TE predictions were expanded into the predictions of drug-pair therapeutic synergy scores together with auxiliary AE estimations (from

DTI commonalities learning). Section 4.1 introduces the subject and defines the formal problem. I detailed the framework design for the problem in section 4.3. Stated in section 4.4, the approach was validated on the specially-collected DTI network-based dataset incorporating reasonable TE and AE information.

In chapter 5, a flexible pre-training-based geometric equivariant framework that can take mutant (binding) complex structures originated from different protein structure modelling tools [e.g., AlphaFold (Jumper et al., 2021) and FoldX (Schymkowitz et al., 2005)] was proposed, for screening candidate protein mutants based on the prediction of binding affinity change starting from wild-type template. Section 5.1 provides the motivation for this work. Section 5.2 validates the proposed method. Section 5.3 further describes details of the overall framework.

In chapter 6, a more general and efficient geometric GRL framework was implemented for broader protein complex candidate screening tasks based on the lightweight MARTINI-based CG-scale protein graph modelling. Within this chapter, it provides comprehensive experimental results on three types of screening tasks over multiple different curated datasets based on various model evaluation settings (section 6.2).

Chapter 7, the final chapter, concludes the work detailed in this thesis and discusses some interesting and promising points that might deserve the further investigation in the future, based on literature.

Chapter 2

Literature Review

This chapter reviews all literature relevant to the concepts identified in chapter 1, highlighting existing research on graph representation learning techniques for solving corresponding tasks at different scales in early drug discovery.

Section 2.1 reviews literature related to GRL methods for biological interaction network-based target identification.

Section 2.2 examines GNN-based heterogeneous network embedding techniques.

Section 2.3 looks at geometric representation learning for screening candidate mutants binding to the protein target.

Section 2.4 provides illustration of general molecular coarse-grained modelling for protein structures.

Finally, section 2.5 summarizes the chapter.

2.1 Graph Representation Learning for Biological Interaction Network-based Target Identification

Before introducing more closely-relevant content for Problem 1 and other Problems (i.e., Research Questions stated in section 1.2), we first briefly introduce the basic definitions of graphs and GRL techniques involved in this thesis. After that, more detailed GRL definitions will be given based on their suitable drug discovery application tasks according to the order of source data scales (from large to small).

2.1.1 Basic Definitions of Graph and Graph Representation Learning

Graph, as a typical non-Euclidean representation of data that contains irregular structure with a varying number of nodes and flexible connection (i.e., edge) patterns, can be used as a universal proxy to describe complex (relationships) systems (Hamilton, 2020), e.g., scientific article citation relationships (Kipf and Welling, 2016), user-item recommendation systems (Wu et al., 2022b), biological interaction networks (Luo et al., 2017), molecular geometric structures (Atz, Grisoni, and Schneider, 2021), etc. Formally, a simplest graph is defined as follows:

$$\mathbb{G} = (V, E) \tag{2.1}$$

where V denotes the entities (i.e, nodes or vertices) set of the graph, and E represents the relations (i.e., connections or edges) between these nodes satisfying $E \subseteq V \times V$. To store the relations or graph structures, the adjacency matrix $\mathbf{A}$ with the basic shape of $|V| \times |V|$ is a common solution where its element $\mathbf{A}_{ij} \neq 0$ represent there is a relation between node i and j in the graph (the weight value depends on the specific task definitions)

In some slightly sophisticated cases, the attributes can be attached to nodes and edges, which will be called as node features and edge features. For example, the encoded atom types and chemical bond types can be leveraged as node and edge features to model a molecular graph, respectively (Nguyen et al., 2021). Additionally, the directions and types of edges can be further distinguished to provide finer information for the relations within the graph (Atz, Grisoni, and Schneider, 2021). Due to the high flexibility and extendability of graphs when modelling different complex systems, the specialized detailed graph definitions for tasks at different data scales are given in Methods section of subsequent chapters.

On top of this, as a mapping function, graph representation learning (GRL, denoted as f here), aims to transform the defined graph into a low-dimensional vector (e.g, embedding for the whole graph) or a set of vectors (e.g., embeddings for the individual nodes) while preserving the graph structure and feature information within the original graph. The derived vectors can be further used for analyses and

predictions for different tasks. The mathematical definition can be given as follows:

$$\mathbf{h}^{emb} = f(\mathbb{G}) \tag{2.2}$$

where $\mathbf{h}^{emb} \in \mathbb{R}^{N \times d}$, N here means the number of generated embeddings, and d is the embedding dimension.

2.1.2 Definitions of Biological Interaction Network-based Target Identification

After giving basic definitions of graph and graph representation learning, under the large-scale coarse-grained candidate screening, the process can be proceeded based on the biological interaction network. Specifically, for the identification of potential drug targets, we can start by constructing a graph in which known drug and target entities as well as other complementary entities (e.g., diseases) are incorporated as the graph nodes, and the available interaction relationships (e.g., interactions between drugs and protein targets validated through wet experiments) are treated as graph edges. Based on the principle that similar drugs may share similar targets and vice versa (Luo et al., 2017), the GRL techniques can be used to explore the interaction relationship similarity between the context drugs and the drug to be inferred (manifested as topology structure similarity around them in the constructed graph), to give the target predictions for the inferred drug based on the embeddings learnt from such similarity. In this case, to provide finer topology structure information for more accurate identifications, relatively more complex graph needs to be defined [i.e., to construct a heterogeneous (interaction) graph], where the node and edge types should be explicitly distinguished to reflect the respective biological importance. The heterogeneous graph is formally defined as follows:

The heterogeneous graph is a graph where multiple types of nodes and edges are explicitly defined following certain constraints. For nodes, they will be partitioned into several disjoint node sets: $V_1 \cup V_2 \cup ... \cup V_n$ where $V_i \bigcap V_j = \varnothing$ and $\forall i \neq j$. For edges, many edges of one type usually only occur between specific types of end

nodes (Hamilton, 2020). For example, a drug-target interaction network can be modelled as a heterogeneous graph incorporating different nodes (e.g., drug and target nodes) and edge types specifying various relationships between these nodes (e.g., the edges with the meaning of "acting on" can be specially designated to indicate the drug nodes imposing interaction effects on the target nodes).

Furthermore, sometimes we need a more structured way to preserve or extract the entities and their relationships for providing more explicit semantic understanding of knowledge contained in the heterogeneous interaction graphs [e.g., for semantic reasoning based on explicit knowledge regulations (Bordes et al., 2013)]. In this context, the heterogeneous graph represented by the aforementioned way could be further transformed into a form of knowledge graph (KG) to achieve more effective reasoning. Specifically, knowledge in KGs is usually expressed as a set of pre-defined triplets in the format of (*head entity*, *relation*, *tail entity*) [e.g., (*drug1*, *acting on*, *target2*)] which can be extracted from traversing the heterogeneous graph. Compared to the heterogeneous graph which is represented based on adjacency matrix and attached with node and edge features emphasizing the strong expression of topology structure complexity (Zitnik, Agrawal, and Leskovec, 2018), KG uses triplets to more consider the semantic relationships between the involved head and tail entities, to enable more structured data storage and reasoning based on logical rules and semantic information (e.g., based on the transitivity or symmetry of defined *relation*) (Yang et al., 2014).

As a contributing reason, due to the difference of the data structure, the GRL methods for directly processing heterogeneous interaction graphs and corresponding KGs, and further, the specific task formulation for target identification could be sightly different, which will be discussed in section 2.1.3

2.1.3 Some Representative Task Formulations and Solutions in Target Identification

Briefly, in many cases, to directly encode the heterogeneous interaction graph containing known DTI relationships (we call heterogeneous DTI graph here), GNNs are a very popular choice, as their more powerful capability to flexibly and efficiently

transform the complex heterogeneous topology structure into low-dimensional embeddings based on the message passing mechanism (Kipf and Welling, 2016) for downstream task usage, compared to some more classical matrix factorization- (Belkin and Niyogi, 2001) and random walk-based (Perozzi, Al-Rfou, and Skiena, 2014) GRL techniques.

Specifically, the message passing-based spatial GNNs (Kipf and Welling, 2016; Veličković et al., 2017; Zitnik, Agrawal, and Leskovec, 2018) prevail as their scalability and efficiency on computing sparse biological networks where known interaction relationships are far less than the unknown ones to be explored (Luo et al., 2017). Mathematically, they can be simply expressed as:

$$\mathbf{h}_i^{l+1} = \phi(\sum_r \sum_{j \in N_r^i} c_r^{ij} W_r^l \mathbf{h}_j^l + c_r^i \mathbf{h}_i^l) \tag{2.3}$$

where $\mathbf{h}_i^{l+1}$ is the updated embedding of node i at the $l+1$ layer of stacked GNN layers, ϕ is the activation function, $r \in R$ represents the current defined edge type, $j \in N_r^i$ gives the direct neighboring nodes of node i connected by edge type r [This can be retrieved from the undirected adjacency matrix (as the symmetry should not affect the relationships between entities in biological networks), e.g., certain pre-defined drug-target relationship edge type in the heterogeneous DTI graph]. c_r^{ij} and c_r^i are the arbitrary normalization coefficients pre-defined in the message passing (acting between node i and j at relation r and individual node i at relation r, respectively). Additionally, $\mathbf{h}_j^l$ and $\mathbf{h}_i^l$ are the embeddings of corresponding nodes at the last layer l (when $l = 0$ they represent the initial node features, which should be designed based on the prior knowledge of current task scenario), and W_r^l denotes the transformation matrix for relation r in layer l.

After the message passing of multiple GNN layers, for each node i, the final obtained node embedding can perceive the information passed from its K-hop neighbors within the graph (K is the total layer number of the defined GNN, and K-hop neighbors represent the nodes that can be reached from the central node i through at maximum K edges), theoretically bringing the fine control for the local topology structure similarity learning to enable the aforementioned similarity-based target

identification (illustrated in section 2.1.2). Next, to complete the target identification, the generated embeddings of drugs and targets can be linked through a mapping function (e.g., a neural network or a prediction function based on cosine similarity) to calculate the score or probability measuring the interaction likelihood of current drug-target pairs. As an example, Peng et al. formulated the target identification as a drug-target pair binary classification task (Peng et al., 2021). After an end-to-end GNN to integrate known relationships between drug, adverse effect, disease, and protein entities in heterogeneous biological networks, an inner product-based prediction function was employed to predict the interaction probability for the unknown drug-target pairs.

In addition to this, to intuitively and explicitly consider relation information during the final target predictions, specialized knowledge graph embedding (KGE) techniques acting on triplets extracted from KGs are an effective solution. Specifically, such KGE methods aim to assign a score for quantifying the likelihood of the *head entity* (i.e., *subject*) and *tail entity* (i.e., *object*) correlating via the *relation* (i.e., *predicate*). Taking a representative translation-based KGE method TransE (Bordes et al., 2013) as an example, it considers a distance-based scoring function where the relation is treated as a translation in the Euclidean embedding space between the head and tail entity embeddings. The scoring function f can be illustrated by the following equation:

$$f = \| \mathbf{h}^{head} + \mathbf{h}^{relation} - \mathbf{h}^{tail} \| \tag{2.4}$$

where $\mathbf{h}^{head}$, $\mathbf{h}^{relation}$, and $\mathbf{h}^{tail}$ denote the learnt embeddings of *head entity*, *relation*, and *tail entity*, respectively. Intuitively, this allows the consideration of additional entity type when inferring the interaction relationships between drugs and target, making the predictions with more conditional controls. For instance, by taking disease types as the *relation* (serving as the extra condition), (Moon et al., 2021) formulated their target identification as a ranking task that ranks the candidate (*drug*, *disease*, *target*) triplets originated from the drug-disease-target (DDT) interaction networks based on the scores provided by the KGE models. This enables screening of

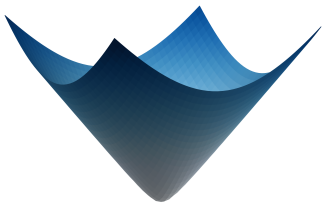


FIGURE 2.1: The geometric shape of the Lorentz feature space. As an isomorphic model of hyperbolic space, it more prevails in GRL techniques due to the numerical stability during model optimization.

targets based on both the drug and disease types. Nevertheless, the demand for further exploiting the topology sub-structures (e.g., hierarchies) of DDT networks preserved by triplets and for better injecting entity priors for handling previously unseen entities (e.g., unseen drugs or targets) deserves deeper exploration.

2.1.4 Hyperbolic Space Embedding

In fact, many complex systems that can be represented as a connected graph share some generic properties, e.g., a high degree of clustering, which can be further interpreted as their hierarchical nature, i.e, small groups of nodes will organize in a hierarchical manner into gradually increasing large groups (Ravasz and Barabási, 2003). The biological interaction graphs could also apply to this regulation (Yue et al., 2023a). When such hierarchies occur in the graph, the representation learning staying in the hyperbolic space has been demonstrated to be an effective solution (Ganea, Bécigneul, and Hofmann, 2018; Chen et al., 2021).

Specifically, the GRL techniques introduced in section 2.1.3 actually work in the Euclidean space, i.e., the generated numerical embeddings keep staying in a feature space in which seeming "obviously true" geometry laws in the world that we live in are postulated, e.g., parallel lines never intersect and the sum of the angles in any triangle is always 180 degrees (McDonald, 2021). In other words, this is a space which is *flat* yet *isotropic*, representing the Gaussian space curvature is zero for the point set within the whole space (Reynolds, 1993).

In this case, Euclidean feature space will grow polynomially, which may limit its capacity to represent hierarchical interaction graphs in which the volume of the hierarchies increases exponentially in proportion to its radius (Yang et al., 2022). While for the hyperbolic space, it is equipped with a constant negative space curvature, which could fit hierarchies with the exponentially increasing capacity and with smaller distortion (Chen et al., 2021). It can be seen as a continuous tree space with explicit geometry constraints, and it has four specific isomorphic models (i.e., representing they can be mapped onto each other through corresponding mappings by which the original properties are preserved) that can be used in GRL methods, including Lorentz model, Poincaré disk model, Poincaré half-plane model, and Klein model (McDonald, 2021). Within these models, the Lorentz model is one of the most popular ones as the numerical stability during model training (Chen et al., 2021). Additionally, there are other types of feature spaces, e.g., spherical space with positive space curvature that is shaped like a globe and has been demonstrated to be effective in capturing cyclic structures (Gu, Sun, and Xu, 2020), while its detailed discussion is out of the scope of the research motivation here.

Based on this theory, how to properly use the hyperbolic space in target identification with explicit consideration of disease types and of biological priors injection for handling unseen entities is the first question to be answered in this thesis. Additionally, more details about the hyperbolic space and the formal mathematical definitions are provided in section 3.2.

2.2 Graph Neural Network-based Heterogeneous Network Embedding for Drug-Drug Therapeutic Effect Prediction

For Problem 2, a coarse-grained screening for the drug combinations that can create stronger synergistic therapeutic synergy effects on certain cancer cell lines (which are usually quantified on the basis of drug-drug pairs), will be solved based on the heterogeneous biological interaction graph embedding. While due to the task formulation difference and the difference on the required source data compared to those in Problem 1, the suitable GRL solutions also differ.

2.2.1 Definitions of Heterogeneous Interaction Graph-based Drug-Drug Therapeutic Effect Prediction

For handling Problem 2 using representation learning methods, the therapeutic effect (TE) prediction task is usually customized as a regression task that takes the representations of a drug-drug pair and of its corresponding cell line as each input sample point, and outputs the real-value therapeutic synergy score supervised by the actual measured value among them (Ryu, Kim, and Lee, 2018). The predicted scores for previously unseen $drug_i - drug_j - cell\ line_k$ combinations can be further analyzed by domain experts for further validation and screening. Besides, because this task requires an accurate prediction close to the actual real-value (i.e., the later one is the regression supervison signal), some ranking-based GRL techniques that focus on giving a relative ranking score mainly for sample sorting are no longer applicable (e.g., the TransE-like approaches illustrated in section 2.1.3).

On top of this, to produce the drug representations for downstream predictions, in addition to directly employing the encoded chemical information of the individual drugs [such as molecular fingerprint, e.g., ECFP (Rogers and Hahn, 2010)] as the embeddings to be further sent to the real-value predictor, the target interaction information extracted from the heterogeneous biological networks could also provide important distinguishing features for improving the corresponding prediction accuracy (Besharatifard and Vafaee, 2024).

2.2.2 Meta-path-based Graph Neural Network for Heterogeneous Biological Graph Learning

As mentioned in section 1.3, we aim to capture the drug-target interaction (DTI) commonalities from the heterogeneous DTI networks within the GRL framework for downstream multi-task learning for TE predictions. To realize this, extracting additional drug representations from GNNs, that can handle DTI patterns that are concisely simulated by pre-defined mata-paths, is a feasible solution.

Specifically, first, meta-paths are a series of combinations of nodes and edges that are extracted from traversing from the heterogeneous interaction graphs following certain pre-defined (meta-path) patterns. For example, we can traverse the graph to

find all combinations following the pattern $drug_i \xrightarrow{interaction_1} target_j \xrightarrow{interaction_2} drug_k$ (in which *drug* and *target* are graph nodes, and *interaction* represent the found edges). These combinations carry the information of $drug_i$ *shares the same* $target_j$ *with* $drug_k$, which can be treated as a proxy of DTI commonality preservation of the relevant drugs for further GRL-based information extraction. Furthermore, multiple groups of such proxies can be pre-defined based on the prior knowledge to cover wider DTI commonality learning.

For the GRL-based information extraction, the vanilla heterogeneous GNNs described in equation (2.3) cannot directly handle meta-paths, each of which is a combination of multiple nodes and edges (rather than two end nodes and an edge between them), therefore the more specialized GNNs should be devised.

It is found that the essence of equation (2.3) is aggregating the neighboring node (i.e., source node) features into each central node (aggregating along each edge type separately) to generate the embedding of the central node (i.e., target node) (Veličković et al., 2017). Keeping this basic principle unchanged, (Wang et al., 2019b) proposed a meta-path-based heterogeneous GNN that, first select the last node in each meta-path as the target node and for each target node find all meta-paths ended with this this node as its meta-path neighbors (i.e., for each meta-path neighbor, only the starting node of the original meta-path is remained as the neighbor, and the intermediate nodes are ignored). The meta-path neighbors are further categorized based on their meta-path types (i.e., meta-path patterns, can be treated as the re-defined new edge types). Based on this, the final each (target) node embedding is acquired based on aggregating its meta-path neighbors under different types separately, followed by a further fusion for these aggregations based on the attention mechanism (Veličković et al., 2017).

Although the effectiveness, this approach might not fully capture the information contained in meta-paths since the intermediate nodes in each meta-path are discarded. (Fu et al., 2020) considered multiple different enhanced aggregation modes that can preserve the information of these intermediate nodes and of potentially their relevant original edges. Nevertheless, how to design proper meta-path patterns to capture the biological importance of DTI commonalities in heterogeneous biological

networks and propose an adaptive GRL framework enabling a more effective meta-path aggregation with the consideration of auxiliary adverse effect learning (i.e., the requirements in section 1.2.2) still needs further investigation.

2.3 Geometric Equivariant Representation Learning for Prediction of Binding Affinity Change Cause by Amino Acid Mutations

2.3.1 Definitions of Graph Neural Network-based Geometric Equivariant Representation Learning

In section 2.1 and section 2.2, the GRL methods are proposed to explore the topological relationship information within a heterogeneous biological interaction network in which a node represents an independent entity, e.g., a small-molecular chemical drug or a macro-molecular biological protein target, to give a relatively coarse-grained screening based on these known entities. Such strategies will fail in some finer early drug discovery tasks, e.g., scoping down the candidate search space of designed similar protein binders (in some cases, these binders only have one amino acid (AA) sequence mutation difference compared with each other) binding to the protein target. This is intuitive as the large-scale 2D biological networks cannot provide enough distinguishing resolution on these candidates.

The tertiary structure of a protein largely determines its function (Petsko and Ringe, 2004). Inspired by this, through capturing the subtle 3D structure difference of these candidates brought by the corresponding AA mutations, the functional difference of the candidates can be further learnt to enable the efficient binder screening. To achieve this, the attention of source data modelling scale will change from large-scale biological networks into small-scale individual molecular structures, and the corresponding GRL-based protein (molecule) learners should also be updated.

Specifically, another distinguishing characteristic for processing protein structures is that, actual protein structures are usually detected from the experimental techniques, e.g., X-ray Crystallography (Ladd, Palmer, and Palmer, 1977), NMR Spectroscopy (Keeler, 2010), and Cryo-Electron Microscopy (Murata and Wolf,

2018), and then deposited in a Protein Data Bank (PDB)-like file format (Burley et al., 2017). The PDB file provides the type and 3D coordinate of every detected atom, which are used to depict the overall structure (i.e., conformation) of this protein in the 3D Euclidean geometric space. Under this circumstance, the GRL methods should evolve to handle the 3D conformation of each protein or protein complex [i.e., a group of two or more associated individual proteins that are bound together by non-covalent interactions to perform a specific biological function (Petsko and Ringe, 2004)].

Further, a protein structure can contain thousands of particles (e.g., atoms at the atom-scale) and irregular chemical bonds between them, which is natural to be represented as a graph (e.g., a graph in which nodes are atoms and chemical bonds between them are edges) and to be further processed by the GNN-based methods. However, due to the introduction of the 3D coordinates, to execute finer representation learning, some specific principles for graph construction and graph representation learning deserve the careful consideration (Stärk et al., 2022). Specifically, compared to the graph defined in equation (2.1), a coordinate matrix $X \in \mathbb{R}^{N\times 3}$ and an initial feature matrix $F \in \mathbb{R}^{N\times d}$ characterizing the node chemical properties are additionally added as follows, above N and d represent the node number and initial feature dimension, respectively.

$$\mathbb{G} = (V, E, X, F) \tag{2.5}$$

In addition, current $\mathbb{G}$ can be treated as a heterogeneous graph to explicitly distinguish the various types of particles and relevant chemical bonds. Given this 3D geometric graph, when encoding it into representations for downstream tasks, an essential physical constraint – the equivariance property that requires regular output changes with the (rigid) change of input 3D coordinates is usually guaranteed. In other words, the overall pose (including position and orientation in 3D Euclidean space) of each actually-detected conformation or conformation generated by some

molecular generation tools could not effectively overlap with each other. From equation (2.3), we can find that the vanilla heterogeneous GNNs are sensitive to this misalignment, i.e., the misalignment will directly influence their output irregularly. This is sometimes sub-optimal as the physical law controlling the dynamics of molecules should not be (irregularly) affected by the pose misalignment (Stärk et al., 2022; Fu et al., 2020). Although the alignment (e.g., performing the protein rigid body transformation like translation and rotation in 3D Euclidean space to make proteins stay in the similar pose) or some augmentation techniques can be used before the GNN processing, they are computational expensive. As an elegant and efficient alternative, heterogeneous GNNs (including their input) can be re-designed to make it directly insensitive to any uncontrollable rigid transformations or regularly reflect the transformation in the output (Han et al., 2022). A simplified mathematical expression for this is as follows, in which x is the individual input data, g is the rigid transformation (e.g., 3D translation or rotation), $f(x)$ can be a GRL model (e.g., a GNN), and $g \cdot x$ or $g \cdot f(x)$ represent that the transformation g is imposed to the later notion (i.e., x and $f(x)$ here).

$$f : X \to Y \qquad f(g \cdot x) = g \cdot f(x) \quad \forall x \in X \tag{2.6}$$

To achieve this, a relatively simple and generic way is to first transform the geometric vectors (e.g., 3D coordinate directly-related information) into invariant scalars (i.e., scalars that are totally insensitive to rigid transformations, e.g., the pairwise geodesic distance between two particles in the 3D Euclidean space). After that, some multi-layer perceptions (MLPs) are attached to control their magnitude, and then they are added to the original directions to obtain the equivariance (Han et al., 2022; Satorras, Hoogeboom, and Welling, 2021). The equivariance property is explicitly considered in many molecular representation learning scenarios, e.g., molecular docking (Stärk et al., 2022), structure generation (Jumper et al., 2021; Xu et al., 2022), and dynamics simulation (Kipf et al., 2018). In our Problem 3 of early drug discovery, the task is formulated as, to devise a specialized equivariant GNN (including its input protein graph design) and corresponding pre-training techniques, which

enable good generalization on perceiving subtle geometric difference between the wild-type (WT) template structure and different mutant structures led by AA mutations, for ultimately screening mutants with stronger binding affinity to the protein target when forming the functional complex (measured by the regressive binding affinity change compared to the WT template binding).

2.4 Coarse-grained Protein Complex Geometric Representation Learning for General Candidate Screening

2.4.1 Definitions of Geometric Graph Representation Learning at The Coarse-grained Particle Scale

Based on the illustration in section 1.2.4, we need to find a set of representation methodologies that can represent a protein complex as a mesoscopic coarse-grained (CG)-scale protein graph between the (microscopic) atom-scale and (macroscopic) residue-scale, which could obsess the advantages of atom-scale representation of finely-described interaction details and of residue-scale representation of computational tractability. At the same time, a set of relevant GRL techniques, including adaptive geometric GNNs and the corresponding pre-training approach for more general machine learning-based candidate screenings, should be provided.

A core challenge is how to give a group of general yet biological-plausible graph construction methods that can produce a fundamental CG-scale heterogeneous 3D geometric node graph preserving essential interaction information for common protein-protein interaction (PPI) systems (to achieve the above purpose). We thought a potential solution is to leverage the mature definitions of CG molecular force field (FF) models used in conventional molecular dynamics (MD) simulation (Monticelli et al., 2008) as the theoretical basis of the framework.

Specifically, MD simulation with (atomistic) FF models aims to simulate a molecular trajectory providing structural and dynamic details that cannot be easily probed experimentally (e.g., revealing the folding kinetics of certain proteins) (Monticelli et

al., 2008; Fu et al., 2022; Lane et al., 2011). During this process, the FF model functions through defining a series of mathematical expressions and parameters for simulating the potential energy and forces between atoms (e.g., the parameters for calculating the non-bonded interaction potential stabilizing two adjacent atoms) within the molecular system, on which the Newton's second law-based motion equations are further solved to update the dynamic trajectory with time (Fu et al., 2022). On top of this, the CG molecular FFs have been proposed to accommodate larger protein systems and longer time scales with lower computation cost for the MD simulation. To realize this, many CG FFs adopt a so-called "top-down" strategy that characterizes the protein systems from a higher-level perspective based on some macroscopic properties, e.g., overall structural properties, thermodynamic quantities, and chemical intuitions (Wang and Gómez-Bombarelli, 2019). More specifically, such CG FFs usually seek for a mapping to separately transform the original atoms and their potential energy mathematical expressions and parameters into the CG interaction sites (as the basic particles) and the expressions and parameters counterparts at the CG-scale based on the empirical choice from widely experimental fitting or the rectification based on machine learning (Souza et al., 2021; Navarro, Majewski, and De Fabritiis, 2023). As a typical example, (Souza et al., 2021) developed MARTINI CG FFs, representing an average of four heavy atoms and their associated hydrogens into one CG bead (and the corresponding potential functions are also provided), parameterized to recover the original thermodynamic properties from comprehensive experimental validations.

Inspired by this, given an extensively tested CG FF that is equipped with explicit definitions of basic particles (e.g., CG beads) and physical or geometric parameters for depicting the inter-particle interaction energies as the theoretical basis, a CG-scale heterogeneous 3D geometric graph could be further designed for GRL techniques to solve the downstream protein (complex) candidate screening problems (e.g., treating the CG FF basic particles as the nodes and the parameters describing the potential energy relationships between adjacent particles as edges). Related to this, some recent works have been proposed to utilize parameters in atomistic FF models as the auxiliary GRL information to predict the property of individual small molecules (Ren et al., 2023). However, how to comprehensively design a specialized

CG-scale GRL framework centered at the efficient CG protein graph construction for general protein (complex) candidate screenings (e.g., binding affinity-, binding affinity change-, and interaction interface type-related screenings) based on proper CG FF selections needs deeper exploration.

2.5 Chapter Summary

This chapter has reviewed relevant works in the GRL techniques especially those based on GNNs for early drug discovery application.

Aiming at several important (molecular-related) candidate screening tasks that require the graph modeling at different source data scales, we formulate them as a series of machine learning (ML)-based GRL tasks, and give the literature review about the recent researches and corresponding supporting techniques. On the top this, we clarify the specific problems that we would like to solve or the drawbacks that we hope to improve in this thesis.

In the following chapters, more detailed illustration, analysis, and discussion will be provided following the order described in section 1.2.

Chapter 3

FLONE: Fully Lorentz network embedding for identification of novel drug targets

This chapter mainly discusses how to explore the hierarchical sub-structures within the heterogeneous drug-disease-target (DDT) networks for drug target identification with explicit consideration of disease types based on hyperbolic space embeddings. Section 3.1 states the importance of the problem and the motivation behind our solution. Section 3.2 introduces our solution in detail. Section 3.3 validates our approach on the curated DDT network datasets to demonstrate the effectiveness of our solution. Finally, section 3.4 summarizes the chapter.

The main contents (including figures) of this chapter have been published in *Bioinformatics Advances* with the DOI: https://doi.org/10.1093/bioadv/vbad066 (Yue et al., 2023a).

3.1 Introduction

Inferring novel drug targets based on computational methods has attracted more attention recently, because it can effectively reduce the time cost in the early stages of drug discovery (Oprea et al., 2011). An important systematic strategy for inferring drug targets is based on analyzing known relationships between drug, target and disease entities in biomedical databases, e.g. DrugBank (Wishart et al., 2006) and

Pharos (Nguyen et al., 2017). The known relationships in these databases are usually constructed as knowledge graphs (KGs) or heterogeneous biological networks and then are analyzed by graph learning algorithms (Bordes et al., 2013; Grover and Leskovec, 2016). For example, (Luo et al., 2017) compiled their dataset DTINet containing drug–disease–target (DDT) relationships from DrugBank, CTD (Davis et al., 2019), HPRD (Prasad et al., 2009) and SIDER (Kuhn et al., 2010) databases, and the network diffusion algorithm and inductive matrix completion strategy were further utilized to infer novel drug–target interactions (DTIs). Other than the network diffusion-based algorithms, many message passing frameworks (e.g., graph neural networks, GNNs) based on various information aggregation mechanisms [e.g., attention mechanism (Yu et al., 2021b)] have been proposed, to learn the multiple topological characteristics of DDT related networks for interaction predictions between drugs and targets (Chu et al., 2022; Peng et al., 2021; Wan et al., 2019; Wang et al., 2022a). In addition, (Ye et al., 2021) generated low-dimensional representations of drugs and targets by integrating their heterogeneous information, extracted from a KG learning method, and their molecular structural information. The final DTIs were predicted by different predictors trained on the produced representations.

However, many methods usually predict putative associations between drugs and targets (i.e. plain drug–target association predictions), ignoring inferring the relationships between diseases and drugs and between diseases and targets. For example, in the entire DTINet, only 530 out of 5603 diseases have modulation relationships with the target *ADRA1A* (*P35348*) of drug *Clozapine* (*DB00363*). In other words, the target and corresponding given drug could be associated with specific diseases. In this case, explicitly considering disease types when inferring drug targets (i.e. DDT association predictions) could bring finer scale virtual screening compared with plain drug–target or drug–disease association predictions, allowing models to predict drug targets under particular disease types directly. Furthermore, we aimed to examine the model performance in a more realistic application scenario. Specifically, if the users want to discover potential targets for the given drug, they usually need to send every DTI combination corresponding to this drug into the trained model, and rank these DTI combinations according to the scores assigned by the

trained model, then the top-ranked DTIs can be selected for further validation experiments (e.g. wet experiments). In this case, the model needs to effectively assign higher ranking scores to the positive target for the given drug and assign relatively lower ranking scores to all other candidate targets for the given drug simultaneously, which is natural to be formulated as a ranking task.

To address these problems, (Moon et al., 2021) formulated the target prediction as a recommendation system ranking task. The ranking task aims to directly assign each candidate target of interest an interaction score for the given drug and disease. The scores should allow positive targets to rank higher than other negative candidate targets. This ranking task formulation better evaluates the model's capability to identify positive target samples from all candidate targets, which better reflects true model performance in actual DTI virtual screening. In addition, based on the similar evaluation idea, (Chen and Li, 2019; Chen and Li, 2020) adopted the ranking task to quantify their model capability on predicting drug–target–disease interactions.

However, the DDTE method proposed in (Moon et al., 2021) could be further improved by capturing the intrinsic hierarchical structure in DDT networks. As shown in Figure 3.1, the entity inter-relationships in a DDT network are hierarchically organized. For example, for a drug node *DB00050* in DTINet, which can be seen as a root node, we know it binds to two target nodes *P30968* and *P22888* directly, with which associate 110 different disease nodes based on the drug–disease and disease–target edges. These DDT relationships essentially form a hierarchical structure, which provides extra topological information about the interactive relationships in the DDT network (Corominas-Murtra et al., 2013). Meanwhile, due to the interconnected triangular relationships between drug, target and disease within the network (i.e. the edges can be constructed between these three types of nodes at the same time) (Walsh, Mohamed, and Nováček, 2020), such hierarchical structures can also be viewed by treating the target or disease nodes as the root node, while treating the other types of associated nodes as its children nodes. Capturing this hierarchical structural property of the DDT network could be helpful to generate more accurate predictions.

Nevertheless, despite the importance, many graph learning methods including DDTE could not fully capture this hierarchical information because they work in

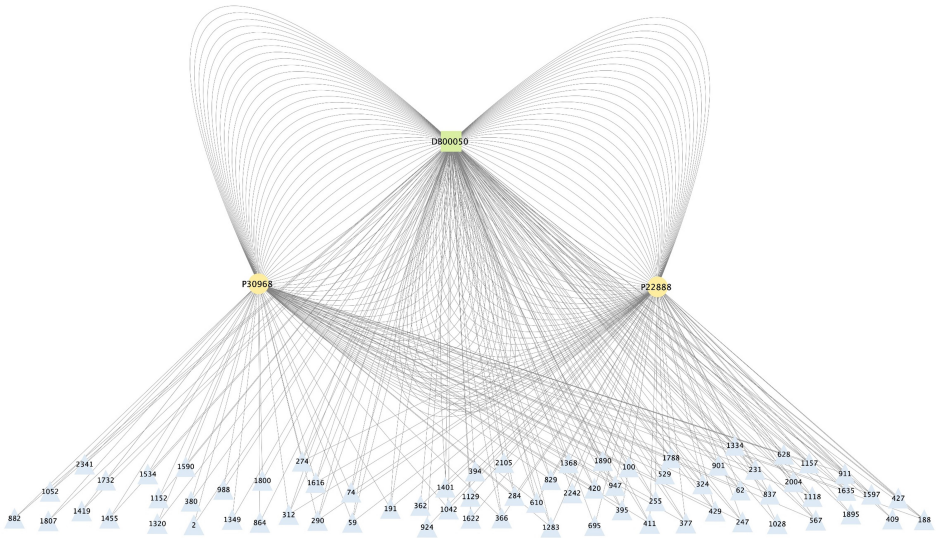


FIGURE 3.1: Illustration of the hierarchies in the DDT network. Taking drug node *DB00050* in DTINet as an example, it can bind to two target nodes *P30968* and *P22888* directly, it can also associate with these two targets through 110 different disease nodes based on the drug–disease and disease–target edges (the labels of disease nodes are disease ids used in our FLONE framework). Thus, starting from *DB00050*, a local hierarchy can be formed (this hierarchy can also be formed starting from the target or disease nodes because these three types of nodes are triangularly interconnected), and such structure can be generalized to other root nodes (in the above example, the root nodes refer to other drugs), forming more complicated implicit hierarchies that reveal the interactive relationships of DDT networks.

Euclidean space. It is known that Euclidean space grows polynomially, limiting its capacity to represent hierarchical networks in which the volume of the hierarchies increases exponentially in proportion to its radius (Yang et al., 2022). In contrast, hyperbolic space, which can be seen as a continuous tree space, could be a better alternative than Euclidean space since it can fit hierarchies with its exponentially increasing capacity and smaller distortion (Chen et al., 2021). Based on this, we hypothesized that, by properly introducing hyperbolic space, graph machine-learning algorithms could better capture the implicit hierarchies in DDT networks, generating more accurate predictions for each candidate target. To test this hypothesis, we formulated the drug–target prediction as a hyperbolic knowledge graph completion (KGC) task explicitly considering disease types, in which drugs and diseases are subjects (i.e., head entities) and predicates (i.e., relations) separately, and targets are the objects (i.e., tail entities) to be completed or predicted. In this way, KGC can be treated as a (*drug*, *disease*, *target*) triplet completion problem, in which the *drug* and *disease* are given and the *target* must be inferred.

To solve this problem, we proposed a proof-of-concept framework, called fully

Lorentz network embedding (FLONE), to identify novel targets associated with the given drug and disease, based on utilizing hyperbolic Lorentzian embeddings to learn implicit structural hierarchies of DDT networks. The main component of FLONE is a hyperbolic similarity calculation module based on a fully Lorentz linear transformation (FuLLiT) (Chen et al., 2021). FuLLiT calculates the Lorentzian distance-based similarity probability score between the hyperbolic embeddings of candidate targets and hyperbolic representation of a given drug under a given disease, which is then used to infer a novel target for the drug–disease combination. Another contribution of our work is that, when identifying the DDT triplets, the capability to handle previously unseen drug and target entities, which are not included by the seen DDT network, is critical in actual application as not every entity at inference time could be linked with known network structures. To extend FLONE to enable processing such types of entities on the hyperbolic space, we devised our hyperbolic drug and target encoders based on the fully Lorentz linear or other optional Lorentz GNN layers, linking these unseen entities with the seen entities using drug and target similarity information.

Within the scope of the aforementioned practical application scenario (i.e., testing the model ranking capability based on identifying positive targets from all candidate targets for the given drug and disease), we conducted extensive experiments on FLONE to test our hypothesis. Our study showed that the DDT scoring or ranking benefit from the introduction of hyperbolic Lorentz space, which supports our hypothesis. In addition to supporting this hypothesis, our results also showed that by fusing the drug structure and target similarity information (as extra domain knowledge), FLONE not only achieved better predictions on DDT triplets related to previously seen drugs and targets, but also could provide accurate predictions on the unseen drugs and targets as well.

3.2 Materials and methods

3.2.1 Datasets

To preserve the heterogeneous DDT networks based on the aforementioned DTINet (Luo et al., 2017) database and an additional BioKG (Walsh, Mohamed, and

Nováček, 2020) database as the proof-of-concept datasets, we first defined the extraction rule of the DDT triplet set. By learning these triplets, KGC models can capture the structural property of original DDT networks (for the subsequent inference). We extracted a triplet ($drug_i$, $disease_k$, $target_j$) [abbreviated as (D_i, $D_k^{'}$, T_j)] if all three of the following edges exist in the original database: (D_i, $D_k^{'}$), (D_i, T_j), ($D_k^{'}$, T_j).

However, in KGC tasks, there could be implicit data leakage caused by very similar predicates, e.g., for two (*subject*, *predicate*, *object*) triplets, (*Birmingham*, *is_in*, *UK*) and (*Birmingham*, *is_located_in*, *UK*), because the predicates *is_in* and *is_located_in* have very similar semantic meaning, thus they correspond to very similar *subject–object* pairs, which causes over-idealistic results when predicting triplets related to *is_in* or *is_located_in*. To avoid this problem, we removed diseases that had >60% drug–target pair similarity with other diseases based on the Jaccard similarity coefficient which can measure the similarity of different (drug–target pair) sets (detailed in appendix A.1). After the screening, 171,597 positive triplets consisting of 535 drugs, 417 targets and 1,160 diseases from DTINet as well as 9,699 positive triplets consisting of 1,128 drugs, 723 targets and 529 diseases from BioKG remained. However, inappropriate data splitting for model evaluation will lead to another type of data leakage, which is detailed in section 3.3.2. In addition, the detailed description of used drug and target similarity information for handling network learning unseen drugs and targets are provided in section 3.2.5.

3.2.2 The basic definition of the Lorentz model

The hyperbolic space is defined as a smooth Riemannian manifold equipped with the constant negative curvature and positive-definite inner product on the tangent space at every point (Yang et al., 2022). There are several isomorphic geometric models of the hyperbolic space: Lorentz model, Poincaré disk model, Poincaré half-plane model and Klein model. In this study, we used the Lorentz model, which is one of the widely used models because of its numerical stability and closed-form computation of geodesics (Chen et al., 2021; McDonald and He, 2022; Sun et al., 2021; Wu et al., 2021; Yu, Visweswaran, and Batmanghelich, 2020)

A n-dimensional Lorentz model with curvature c [please note that c could be set to trainable or fixed (usually set to -1 for an easy optimization) (He, Yang, and

Ying, 2024), here we used fixed -1 for a clear expression] is defined as $\mathbb{L}_c^n = (\mathcal{L}^n, g_x^c)$, $g_x^c = \eta$ is the Riemannian metric tensor satisfying $\eta = I$ (I is the $(n+1)$-dimension diagonal matrix) except $\eta_{0,0} = -1$, and $\mathcal{L}^n$ represents the manifold defined by the point set in $\mathbb{L}_c^n$:

$$\mathcal{L}^n := \{x \in \mathbb{R}^{n+1} : \langle x, x \rangle_{\mathcal{L}} = 1/c, x_t > 0\} \tag{3.1}$$

$$\langle x, y \rangle_{\mathcal{L}} := x^T g_x^c y = -x_t y_t + \mathbf{x_s}^T y_s \tag{3.2}$$

where $\langle ..., ... \rangle_{\mathcal{L}}$ is the Lorentzian inner product. Each point x in $\mathbb{L}_c^n$ is expressed as a concatenation $[x_t, \mathbf{x_s}]$, where $x_t \in \mathbb{R}$ (referred as the time dimension), $\mathbf{x_s} \in \mathbb{R}^n$ (referred as the spatial dimension), and the origin $\mathcal{O}$ of the Lorentz model is defined as $(\sqrt{-1/c}, 0, ..., 0)$. In other words, x represents the space coordinate point in the defined n-dimensional Lorentz space, all drugs and targets will be embedded into this Lorentz space, and the allocated coordinate points or positions of drugs and targets can be seen as their feature embeddings. The n-dimensional Lorentz space consists 1 temporal dimension and n spatial dimensions, which are used to depict the coordinate position or embedding feature on each dimension here. For further understanding, we suggested referring (Yang et al., 2022) for a more detailed description.

Besides, for every point $x \in \mathbb{L}_c^n$, it is equipped with an orthogonal space (i.e., tangent space) of $\mathbb{L}_c^n$ (at x), which is the first-order approximation of $\mathcal{L}^n$ around x (Yang et al., 2022), and is formally defined as $\mathcal{T}_x\mathbb{L}_c^n := \{z \in \mathbb{R}^{n+1} : \langle z, x \rangle_{\mathcal{L}} = 0\}$, where z is the point set of this tangent space.

3.2.3 Description of the FLONE method

Based on the defined Lorentz model, we proposed FLONE to solve the drug-disease-target [i.e., (*drug*, *disease*, *target*)] triplet target entity completion problem. Specifically, for each drug-disease-target triplet in the extracted triplet set, FLONE treats the drug, disease, and target as the subject, predicate (relation), and object, respectively. The task is, given a drug-disease combination, FLONE will assign a similarity

score to every candidate target (i.e., object entity) in the DDT network, which indicates the distance-based similarity between the target and given drug and disease. FLONE then uses the similarity scores to rank these targets, for identifying high-confidence targets for the drug-disease combination.

A high-quality model would assign high similarity scores to all of the defined drug-disease-target triplets that have been extracted from the DDT network, and low similarities for all other drug-disease-target combinations. A researcher investigating drug repurposing would provide the drug and disease of interest and will have a ranked list of targets returned for further investigation.

The illustration of FLONE is shown in figure 3.2. After the extraction of triplets from the heterogeneous DDT network, we presented FLONE as an end-to-end framework consisting of three major components:

- FuLLiT, a feed-forward neural network operating within Lorentzian space, which was proposed to increase the performance of the Euclidean translation-based KGC method (more explanation about the translation-based methods are in section 2.1.3) in our task. FuLLiT treats a (disease) predicate or relation as a trainable translation offset from the (drug) subject to the (target) object entity embeddings. It can take two n-dimensional Lorentzian vectors, representing a drug embedding and a target embedding, along with a disease index as inputs, and outputs a similarity score for the drug-disease-target triplet. FuLLiT is detailed in section 3.2.4.

- Hyperbolic drug encoder, which is responsible for taking drug chemical features to produce a hyperbolic Lorentzian embedding for a drug. It can inject drug domain knowledge into FuLLiT for handling previously unseen drugs at inference time (explained in more detail in section 3.2.5).

- Hyperbolic target encoder, which, likewise, produces hyperbolic Lorentzian embeddings for all candidate targets, to fuse target domain knowledge into FuLLiT for handling unseen targets (detailed in section 3.2.5).

In addition, all important hyperbolic computation operations of FLONE are fully Lorentzian instead of a hybrid mode (Chen et al., 2021). This has the advantage that

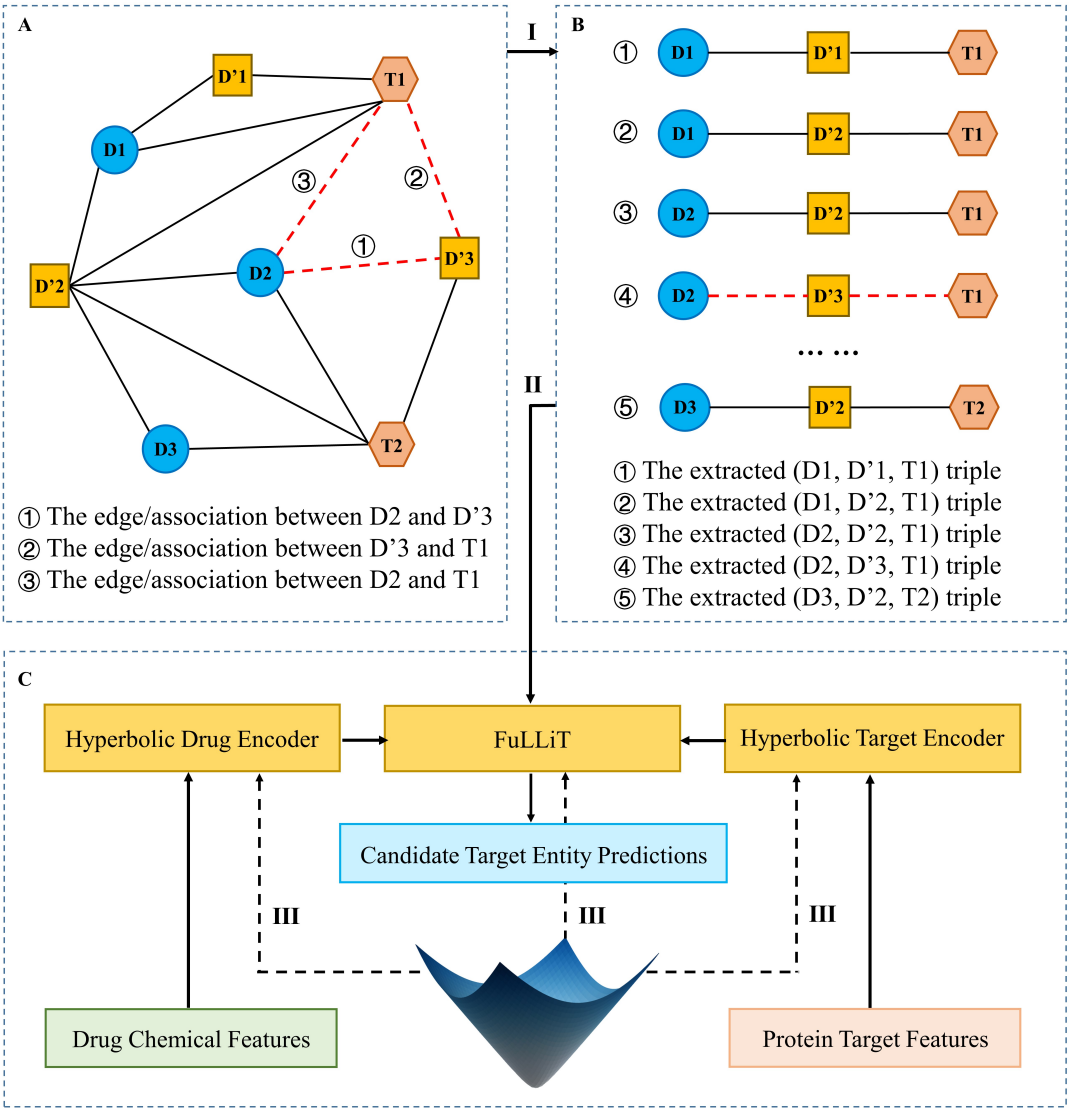


FIGURE 3.2: Illustration of FLONE. (A) Construct a heterogeneous drug-disease-target network. (B) Decompose this network to extract the drug-disease-target triple(t) set. (C) Input the Euclidean drug and target (similarity) domain knowledge into the hyperbolic drug and target encoders respectively, to generate hyperbolic (Lorentzian) drug and target embeddings (or using self-contained drug and target embeddings in FuLLiT directly). Next, the hyperbolic drug and target embeddings are used by FuLLiT, in which hyperbolic drug embeddings are transformed by the disease translation representations, and then the Lorentzian distance-based similarity between the transformed drug embeddings and hyperbolic target embeddings are calculated. After training, FLONE is able to rank every candidate target entity for the given drug and disease.

there is no need to expensively map between the Lorentz space and its tangent space (McDonald and He, 2022).

3.2.4 Description of FuLLiT

Based on the fully Lorentz linear layer, FuLLiT was developed, which includes three components: 1) a hyperbolic triplet decoder, 2) a self-contained drug embedding look-up table, 3) and a self-contained target embedding look-up table. The first component is to calculate the similarity score between the target and given drug and disease for each drug-disease-target triplet, and the latter two are for providing corresponding hyperbolic embeddings without drug or target similarity information (i.e., only utilizing structural information of DDT networks). These three components form the backbone of FLONE that can deal with cases where external domain knowledge is unavailable.

The general form of the fully Lorentz linear layer $FLLinear_{n,m}(x)$ (contained in FuLLiT) is defined as equation (3.3), which could ensure a linear transformation to map $x \in \mathbb{L}_c^n$ ($x \in \mathbb{R}^{n+1}$) to $y \in \mathbb{L}_c^m$ ($y \in \mathbb{R}^{m+1}$) (i.e., the coordinates of input x and output y of the transformation are guaranteed to stay in the respective Lorentz space):

$$\begin{aligned} y &= FLLinear_{n,m}(x) \\ &= \frac{\sqrt{(\lambda\sigma(v^T x + b) + \epsilon)^2 + 1/c}}{\|W(dropout(x))\|} W(dropout(x)) \end{aligned} \tag{3.3}$$

where λ is a fixed hyper-parameter to control the numerical scale of the time dimension, σ is the sigmoid function, $v \in \mathbb{R}^{n+1}$ and $W \in \mathbb{R}^{m\times(n+1)}$ are trainable weights of the overall linear transformation matrix $M = \left[\frac{v^T}{W}\right]$ ($\left[\frac{\cdots}{\cdots}\right]$ represents the tensor concatenation) in this linear layer. Besides, b is the trainable bias, ϵ is a fixed value larger than $\sqrt{\frac{-1}{c}}$ to ensure $\sqrt{(\lambda\sigma(v^T x + b) + \epsilon)^2 + 1/c}$ larger than 0. In FLONE, we adopted the same dimension for all embeddings, in this case, n is equal to m in every intermediate $FLLinear_{n,m}(x)$ layer.

For the hyperbolic triplet decoder of FuLLiT, assuming embedding all entities into a n-dimensional Lorentz space, the following steps are included to calculate the

similarity scores (from 0 to 1). Take an example of predicting the score for T_j under $(D_i, D_k^{'})$, five sets of parameters are needed, including corresponding drug and target embeddings, disease type specific translation offset (for $D_k^{'}$), and real-value drug and target biases. Specifically, we first obtained the hyperbolic embedding $x_{D_i} \in \mathbb{L}_c^n$ and $x_{T_j} \in \mathbb{L}_c^n$ of D_i and T_j from the self-contained embedding look-up tables (i.e., the tables that store the corresponding type of embeddings through entity indices). Then, $x_{D_i} \in \mathbb{L}_c^n$ was transformed by the disease type specific translation offset $FLLinear_{n,n}(x)$ (for $D_k^{'}$), to obtain the translation $x_{D_i - D_k^{'}} \in \mathbb{L}_c^n$. Then the similarity score p can be calculated as (Balazevic, Allen, and Hospedales, 2019; Chen et al., 2021):

$$p(D_i, D_k^{'}, T_j) = \sigma(-d_L^2(x_{D_i - D_k^{'}}, x_{T_j}) + b_{D_i} + b_{T_j} + \phi) \tag{3.4}$$

$$d_{\mathcal{L}}^2(x_{D_i - D_k^{'}}, x_{T_j}) = \frac{2}{c} - 2\langle x_{D_i - D_k^{'}}, x_{T_j} \rangle_{\mathcal{L}} \tag{3.5}$$

where $d_{\mathcal{L}}^2(x_{D_i - D_k^{'}}, x_{T_j})$ represents the squared Lorentzian distance measuring the similarity between the given drug-disease combination and corresponding target (Law et al., 2019; Ratcliffe, 2006), b_{D_i} and b_{T_j} are the drug and target type specific biases, respectively. ϕ is the margin hyper-parameter, and σ is the sigmoid function.

As for the above self-contained drug and target embedding look-up tables, they are essentially trainable matrices formally defined as $drug_lookup$ and $target_lookup$ with the shapes of $(drug\ number, n)$ and $(target\ number, n)$, in which n is the overall feature dimension of the Lorentz model, drug number and target number are the number of all drugs of interest and all candidate targets, respectively:

$$x_{D_i} \in \mathbb{L}_c^n \leftarrow drug_lookup(D_i) \tag{3.6}$$

$$x_{T_j} \in \mathbb{L}_c^n \leftarrow target_lookup(T_j) \tag{3.7}$$

Specifically, the shape of the look-up tables are determined based on the number

of drugs and targets under which we would like to explore the potential drug targets. They can be trained together with other sets of parameters, to produce the required hyperbolic embeddings based on learning topological structural in-formation of DDT networks. Besides, the look-up tables can also be created by hyperbolic drug and target encoders (detailed in section 3.2.5) in an end-to-end way (all sets of parameters are learnt simultaneously based on known triplets extracted from DDT networks), for injecting external domain knowledge to handle the unseen drugs and targets.

As a comparison, the Euclidean translation-based method DDTE (Moon et al., 2021) mentioned in section 3.1, uses vectors as the disease translation offsets. Specifically, it used vector-based disease offset $V_{D'_k}$ to calculate the translated drug embedding under the given disease (i.e., $e_{D_i} + V_{D'_k}$, e_{D_i} is the Euclidean drug embedding). While for FuLLiT, it was calculated based on the matrix-based offset $FLLinear_{n,n}(x_{D_i})$. Except for specialized hyperbolic operations, another two main differences between DDTE and FuLLiT or FLONE are that compared with FLONE, when calculating the similarity score, there are no drug and target specific biases in equation (3.4) for DDTE, and DDTE cannot consider external domain knowledge (i.e., only providing Euclidean self-contained drug and target look-up tables).

3.2.5 The hyperbolic encoders for fusing drug and target similarity information

To investigate whether the triplet target entity completion benefits from the fusion of domain knowledge, we extended the FuLLiT, by replacing the self-contained drug and target embedding look-up tables with the tables generated by corresponding hyperbolic encoders, for encoding and injecting external domain knowledge into embeddings, and the size of these embeddings is same as that in the self-contained look-up tables.

Specifically, we tried to fuse different domain knowledge of drug-drug and target-target similarities by utilizing hyperbolic drug and target encoders. Intuitively, this could bring similar hyperbolic embeddings for involved similar drugs or targets. Based on the assumption "similar drugs may share similar targets and

vice versa" (Luo et al., 2017), this enables the further predictive performance improvement of FLONE.

For the drugs of interest, we computed Extended Connectivity Fingerprints of diameter 6 (ECFP6), a circular topological fingerprint commonly used in drug discovery (Rogers and Hahn, 2010), as the input of the hyperbolic drug encoder. For involved targets, we provided two types of optional target features to be encoded by the hyperbolic target encoder: graph-like features (GO-PPI network) and sequence-like features [target sequence similarity, i.e., protein similarity scores based on primary sequences of target proteins after 0-1 normalization (Luo et al., 2017)]. A detailed description of constructing the GO-PPI network is in appendix A.2.

Since all numerical operations in FLONE are defined on Lorentz space, these (Euclidean) drug and target features must first be mapped to vectors on the Lorentzian manifold $\mathcal{L}^n := \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle_{\mathcal{L}} = 1/c, x_t > 0\}$. To this end, the tangent space of Lorentz space $\mathcal{T}_x\mathbb{L}_c^n := \{z \in \mathbb{R}^{n+1} : \langle z, x\rangle_{\mathcal{L}} = 0\}$ is needed for the mapping. Specifically, for the hyperbolic drug encoder, it takes the (Euclidean) ECFP6 as the input. Take encoding ECFP6 of D_i (termed as $ECFP_{D_i}$) as an example, first, $ECFP_{D_i}$ is concatenated with 0 to create $(0, ECFP_{D_i})$, to map $ECFP_{D_i}$ into the tangent space of origin of Lorentz space $\mathcal{O} = (\sqrt{-1/c}, 0, ..., 0)$. Because according to $\mathcal{T}_x\mathbb{L}_c^n := \{z \in \mathbb{R}^{n+1} : \langle z, x\rangle_{\mathcal{L}} = 0\}$ and $\langle (0, ECFP_{D_i}), \mathcal{O}\rangle_{\mathcal{L}} = 0$, $(0, ECFP_{D_i})$ is in the tangent space of $\mathcal{O}$. Then, $(0, ECFP_{D_i})$ is further mapped into the Lorentz space through the exponential map function defined as follows, where $z \in \mathcal{T}_x\mathbb{L}_c^n$:

$$exp_x^c(z) = cosh(\sqrt{|c|}\|z\|_{\mathcal{L}})x + z\frac{sinh(\sqrt{|c|}\|z\|_{\mathcal{L}}}{\sqrt{|c|}\|z\|_{\mathcal{L}}} \tag{3.8}$$

$$\|z\|_{\mathcal{L}} = \sqrt{\langle z, z\rangle_{\mathcal{L}}} \tag{3.9}$$

In this way, the Lorentzian ECFP6, i.e., $exp_{\mathcal{O}}^c((0, ECFP_{D_i}))$ can be generated, and then $exp_{\mathcal{O}}^c((0, ECFP_{D_i}))$ is sent to $FLLinear_{m,n}(x)$ specifically for encoding Lorentzian ECFP6, to reduce its dimension to the unified hidden dimension mentioned in section 3.2.4. Furthermore, all Lorentzian ECFP6 (after dimension reduction) of involved drugs constitutes the hyperbolic drug embedding look-up table (through drug entity indices). As an analogy (above procedure can be used to

map any arbitrary Euclidean vector to a Lorentzian vector), based on the similar procedure with a $FLLinear_{m,n}(x)$ specifically for target sequence similarity, the hyperbolic target encoder (for sequence-like features) can be generated. In addition, the description of generating the hyperbolic target embedding look-up table based on the (graph-like) GO-PPI network encoded by a Lorentz GNN is provided in appendix A.2.

3.2.6 Model training and optimization

Similar to the Euclidean translation-based KGC models, negative sampling (Bordes et al., 2013) was used to train the FLONE. Specifically, during the training phase, for each positive or known triplet $(D_i, D_k^{'}, T_j)$, the negative triplet was sampled by randomly replacing T_j with other $T_{j'}$ in the target entity set and ensuring that the generated negative triplets were not in the extracted known drug-disease-target triple set. The loss function used for optimizing all sets of parameters in FLONE was the binary cross entropy (BCE) loss defined as follows:

$$L = -\frac{1}{N}\sum^{N}(logp(D_i, D_k^{'}, T_j)) + \sum^{N^{'}} log(1 - p(D_i, D_k^{'}, T_{j'})) \tag{3.10}$$

where N is the positive or known triplet number in the training set and $N^{'}$ is the number of negative triplets generated for each known triplet. Besides, we adopted the Riemannian Adam as the optimizer of FLONE, which is the counterpart of Adam defined in Euclidean space (Bécigneul and Ganea, 2018; Kochurov, Karimov, and Kozlukov, 2020).

3.3 Results and discussion

3.3.1 Hyperbolicity of the DDT networks and scalability analysis

To test whether the resulting heterogeneous DDT networks (equivalent to the extracted triplet sets) exhibit a hierarchical structure for demonstrating our hypothesis, we calculated their Gromovs hyperbolicity δ (Chami et al., 2019; Gromov, 1987),

which measures how hierarchical the network is. The lower δ, the more implicit hierarchies the network has, and δ of completely tree-like structures is 0. Moreover, for common hierarchical benchmark datasets, e.g., Human PPI and Airport, the δ is about 1, and for the standard (non-hierarchical) benchmark, e.g., Cora, the δ is 11 (Chami et al., 2019). While the calculated δ values of the extracted DTINet and BioKG DDT networks were both 1.5, which indicated that these networks do possess the implicit hierarchies, making them theoretically feasible for hyperbolic space embedding.

The model complexity of FLONE is $O(dn_e + n_e + n_r d^2)$, in which d, n_e, n_r are the embedding dimension, (drug and target) entity number, and (disease) relation number, respectively. Specifically, the complexity of either self-contained look-up tables or hyperbolic encoders based on $FLLinear_{m,n}(x)$ ($O(dn_e)$) is proportional to d and n_e and irrelevant to n_r, as the matrix contained in them is independent of disease types. Besides, as demonstrated by (Chen et al., 2021), the fully Lorentzian-based KGC algorithm can be effectively extended to KG triplet datasets with over 40,000 entity number and hundreds of relation types (Dettmers et al., 2018; Toutanova and Chen, 2015). To the best of our knowledge, this scale can satisfy handling most of the precisely curated DDT triplet sets.

3.3.2 Model evaluation settings

Except for the data leakage mentioned in section 3.2.1, the common model evaluation setting, i.e., randomly splitting the drug-disease-target triplet set into training, validation, and test sets could also cause over-idealistic results in our task. This is because each defined known (D_i, $D_k^{'}$, T_j) triplet not only contains the associations (D_i, $D_k^{'}$) and ($D_k^{'}$, T_j), but also includes (D_i, T_j) (i.e., constituting a triangular internode sub-structure), and ignoring (D_i, T_j) in data splitting will lead to extra data leakage. For example, if (D_i, $D_{k1}^{'}$, T_j) and (D_i, $D_{k2}^{'}$, T_j) are allocated into training and test sets separately, after training, when inferring the target for (D_i, $D_{k2}^{'}$), T_j tends to be chosen more easily, as the model "has already seen" the unnecessary implicit association information between D_i and T_j [i.e., (D_i, T_j)] during the training phase.

To avoid this pitfall, we split these triplets into training, validation and test sets

based on drug-target pairs. In other words, the known triplets with the same drug-target pair were put into the same set, as such all test targets for a given drug were not directly seen by the model during training. In this case, no consideration was given to ensure a complete coverage of drugs and targets in the training set, and so it was possible for drugs and targets to exist in either the validation or test set, but not in the training set (the related statistics are in appendix A.3), which also increased the difficulty of target identifications.

Based on this setting, the drug-disease-target triplets corresponding to 60%: 20%: 20% of all drug-target pair varieties were divided into training, validation, and test sets, respectively. This procedure was repeated five times independently, for each time, before splitting data, the whole drug-disease-target triplet set was randomly shuffled to make different drug-target pair varieties enter each set. We computed and reported the average evaluation metrics over the five independent repeats.

To evaluate the model's predictive performance in the scenario of explicitly considering disease types and directly ranking all candidate targets, we adopted the standard evaluation metrics used in recommendation system ranking tasks, including Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10 (Moon et al., 2021). Among these, MRR was chosen as the main metric, because it can better evaluate the model's ability to assign the positive target a ranking score that is distinguishable from other candidate targets (under the given drug and disease). MRR is calculated as follows:

$$MRR = \frac{1}{|N_{test}|} \sum_{i=1}^{|N_{test}|} \frac{1}{rank_i} \tag{3.11}$$

where N_{test} is the test sample set, and $rank_i$ is the score ranking of the true target entity among all targets of interest for the i_{th} test sample. Hits@K represents the percentage of the true target entities that appear within the top K positions of overall candidate ranking during the test phase.

3.3.3 Experiments without external domain knowledge injection

The main objective of our experiments is to test whether properly introducing hyperbolic space can improve the performance of Euclidean (translation-based) KGC methods in our task. To this end, we compared our method with the representative Euclidean (translation-based) method DDTE detailed in section 3.2.4 based on the aforementioned data splitting. To eliminate the uncontrolled influence brought by different external domain knowledge for a fair comparison, we first chose FuLLiT (termed as FLONE$_{base}$, later these two names will be used interchangeably), which only uses self-contained drug and target embedding look-up tables (the performance of FLONE with the hyperbolic encoder under the same experimental settings and data splitting was discussed in section 3.3.4). The two additional algorithm variants were also considered. The first variant was named as DDTE$_{bias}$, in which the drug and target type specific biases mentioned in equation (3.4) were added into DDTE. The second variant was the fully Euclidean counterpart of FLONE$_{base}$, termed as FEC-FLONE$_{base}$: on the top of DDTE$_{bias}$, the vector-based disease offset was replaced by the matrix-based offset (i.e., using $ELinear_{n,n}(e_{D_i})$ as the offset where $ELinear_{n,n}$ and e_{D_i} represent the Euclidean linear layer and Euclidean drug embedding separately). This controls for the internal representation of diseases by representing diseases as translation matrices in both algorithms.

To conduct more comprehensive comparison, we also considered two other Euclidean KGC methods ConvE (Convolution-based) (Dettmers et al., 2018) and DistMult (bilinear product-based) (Yang et al., 2014) as well as another hyperbolic method MuRP (Poincaré space-based) (Balazevic, Allen, and Hospedales, 2019). Further, we did experiments based on two representative embedding dimensions: 16 and 128. 16 and 128 are commonly selected, representative dimensions for hyperbolic and Euclidean embedding in downstream tasks, respectively (Chen et al., 2021; Moon et al., 2021). Theoretically, hyperbolic embedding loses less information than Euclidean embedding when the embedding dimension is small. Besides, the Adam optimizer (Kingma, 2014) was adopted for all Euclidean-based models.

[1]The bold data indicates the best result under current evaluation metric and dataset.
[2]The bold data indicates the best result under current evaluation metric and dataset.

TABLE 3.1: Comparison results of involved methods under the 16 embedding dimension.

DTINet	MRR	Hits@1	Hits@3	Hits@10
FLONE$_{base}$	**0.3806**	**0.3301**	**0.3994**	0.4685
FEC-FLONE$_{base}$	0.3608	0.3174	0.3680	0.4484
DDTE$_{bias}$ (Moon et al., 2021)	0.3134	0.2315	0.3520	0.4651
DDTE (Moon et al., 2021)	0.3049	0.2123	0.3518	**0.4790**
MuRP (Balazevic, Allen, and Hospedales, 2019)	0.3234	0.2463	0.3605	0.4636
ConvE (Dettmers et al., 2018)	0.3160	0.2362	0.3583	0.4648
DistMult (Yang et al., 2014)	0.3014	0.2148	0.3387	0.4739
BioKG	MRR	Hits@1	Hits@3	Hits@10
FLONE$_{base}$	**0.4706**	**0.3786**	**0.5224**	**0.6452**
FEC-FLONE$_{base}$	0.3594	0.3049	0.3797	0.4600
DDTE$_{bias}$ (Moon et al., 2021)	0.4222	0.3302	0.4653	0.5992
DDTE (Moon et al., 2021)	0.3545	0.2417	0.4162	0.5586
MuRP (Balazevic, Allen, and Hospedales, 2019)	0.4358	0.3373	0.4911	0.6146
ConvE (Dettmers et al., 2018)	0.3737	0.2815	0.4182	0.5507
DistMult (Yang et al., 2014)	0.2877	0.1986	0.3310	0.4518

[1]

TABLE 3.2: Comparison results of involved methods under the 128 embedding dimension.

DTINet	MRR	Hits@1	Hits@3	Hits@10
FLONE$_{base}$	**0.4335**	**0.3936**	**0.4428**	0.5054
FEC-FLONE$_{base}$	0.4098	0.3705	0.4139	0.4907
DDTE$_{bias}$ (Moon et al., 2021)	0.3893	0.3232	0.4165	**0.5125**
DDTE (Moon et al., 2021)	0.3139	0.2186	0.3635	0.4997
MuRP (Balazevic, Allen, and Hospedales, 2019)	0.3310	0.2502	0.3699	0.4823
ConvE (Dettmers et al., 2018)	0.3885	0.3389	0.4090	0.4702
DistMult (Yang et al., 2014)	0.3555	0.3000	0.3780	0.4536
BioKG	MRR	Hits@1	Hits@3	Hits@10
FLONE$_{base}$	**0.5027**	**0.4116**	**0.5495**	**0.6758**
FEC-FLONE$_{base}$	0.4549	0.3790	0.4914	0.5960
DDTE$_{bias}$ (Moon et al., 2021)	0.4754	0.3780	0.5215	0.6671
DDTE (Moon et al., 2021)	0.3806	0.2549	0.4576	0.6071
MuRP (Balazevic, Allen, and Hospedales, 2019)	0.4750	0.3749	0.5323	0.6586
ConvE (Dettmers et al., 2018)	0.4641	0.3781	0.5090	0.6296
DistMult (Yang et al., 2014)	0.3863	0.3176	0.4198	0.5108

The experimental results are in table 3.1 and table 3.2. Both dimensions suggest that FLONE$_{base}$ achieved overall better performance compared with all involved Euclidean-based models across both embedding dimensions and on both the DTINet and BioKG datasets. Only under Hits@10, the sub-optimal performance was obtained on DTINet. Compared with the second-best model (based on the main metric MRR), FLONE$_{base}$ obtained 5.5% and 5.8% (on DTINet) and 8.0% and 5.7% (on BioKG) performance margins on the 16-dimension and 128-dimension, respectively. The above clearly verified the effectiveness of FLONE and our hypothesis. Interestingly, MuRP did not produce very close results with our Lorentz space-based method due to numerical instability of the Poincaré model, suggesting our decision to use the Lorentz model.

3.3.4 Fusing domain knowledge for FLONE

After demonstrating our basic hypothesis, following the same experimental settings and data splitting as the last section, we investigated the effectiveness of fusing domain knowledge for FLONE. As described in section 3.3.2, current data splitting is based on a challenging scenario where previously unseen drugs and targets exist. It is difficult for FuLLiT and DDTE to handle the included unseen drugs and targets because what they leverage is only known structural information of the DDT network. Intuitively, the fusion of similarity domain knowledge of involved drugs and targets could be helpful to improve the prediction accuracy of FuLLiT. To demonstrate this, based on the larger and more complex DTINet DDT network with complete similarity data, the FLONE variants with different hyperbolic drug and target encoder combinations for FuLLiT (i.e., FLONE$_{ECFP-SEQ}$, FLONE$_{ECFP-NET}$, FLONE$_{ECFP-None}$, FLONE$_{None-SEQ}$, and FLONE$_{None-NET}$), were added into performance comparison. The two suffixes (first: for drugs, second: for targets) of the variant name represent the use of the corresponding encoders ($ECFP$: hyperbolic ECFP6-based drug encoder, SEQ: hyperbolic target sequence similarity-based target encoder, NET: hyperbolic GO-PPI network GNN-based target encoder, None: using the original self-contained embedding look-up table without domain knowledge injection). Additionally, we compared two representative network-based target prediction methods NeoDTI (Wan et al., 2019) and KGE_RF (Ye et al., 2021) which

were trained based on a standard 1:1 sampling of positive and negative samples (the detailed experimental setup of these two methods are in appendix A.4). Since the final embedding dimension of KGE_RF depends on the selected drug and target structural information dimension, we adopted their default basic hyper-parameters in the original paper (Ye et al., 2021).

Table 3.3 gives the predictive performance of all of the aforementioned algorithms under the experimental setting described in section 3.3.2. From the results we first found that, compared with the other FLONE variants, FLONE$_{ECFP-SEQ}$ had the best performance. Under both 16 and 128 dimensions, it obtained 24.5% and 30.2% performance improvements on MRR compared with FLONE$_{base}$ (i.e., FuLLiT). This can demonstrate the importance of similarity domain knowledge in current scenario where unseen drugs and targets occur. Interestingly, FLONE$_{ECFP-SEQ}$ and FLONE$_{None-SEQ}$ were better than FLONE$_{ECFP-NET}$ and FLONE$_{None-NET}$, respectively. We thought the contributing reason was that, the graph-based target encoder brought unnecessary complexity, causing negative effects on the learning process, making it worse than the sequence-based target encoder. Besides, FLONE$_{ECFP-None}$ was better than FLONE$_{None-SEQ}$ and FLONE$_{None-NET}$, indicating that ECFP6 could be more effective compared with the provided two types of target features in our task. Meanwhile, FLONE$_{ECFP-SEQ}$ also outperformed the involved NeoDTI and KGE_RF algorithms, which further demonstrated the effectiveness of our framework. On top of this, we also provided the experiments about fusing similarity domain knowledge into the Euclidean-based models in appendix A.5.

3.3.5 Extra ablation study

Based on the conclusion from section 3.3.4, to confirm that the performance gain seen in FLONE having drug and target encoders compared with the one without the encoders, is not solely due to the later one's inability to handle unseen drugs and targets, we devised an extra ablation study: We still kept the same experimental settings, to run FuLLiT (FLONE$_{base}$) and FLONE$_{ECFP-SEQ}$ again based on the DDT network screened from the DTINet dataset. The only difference here was that we divided the original test set of each independent repeat into three parts, and computed

[3]The bold data indicates the best result under current evaluation metric.

TABLE 3.3: Comparison results of involved methods under 16 and 128 embedding dimensions based on the DTINet dataset.

Methods	MRR	Hits@1	Hits@3	Hits@10
FLONE$_{ECFP-SEQ}$ [dim=16]	**0.4740**	**0.4121**	**0.4945**	**0.5946**
FLONE$_{ECFP-NET}$ [dim=16]	0.4475	0.3841	0.4776	0.5599
FLONE$_{ECFP-None}$ [dim=16]	0.4322	0.3750	0.4568	0.5347
FLONE$_{None-SEQ}$ [dim=16]	0.4068	0.3461	0.4233	0.5260
FLONE$_{None-NET}$ [dim=16]	0.3865	0.3325	0.4102	0.4804
FLONE$_{base}$ [dim=16]	0.3806	0.3301	0.3994	0.4685
NeoDTI [dim=16]	0.1464	0.0539	0.1563	0.3245
FLONE$_{ECFP-SEQ}$ [dim=128]	**0.5644**	**0.5139**	**0.5793**	**0.6564**
FLONE$_{ECFP-NET}$ [dim=128]	0.5084	0.4606	0.5308	0.5911
FLONE$_{ECFP-None}$ [dim=128]	0.5033	0.4571	0.5216	0.5912
FLONE$_{None-SEQ}$ [dim=128]	0.4820	0.4410	0.4892	0.5614
FLONE$_{None-NET}$ [dim=128]	0.4363	0.3923	0.4510	0.5121
FLONE$_{base}$ [dim=128]	0.4335	0.3936	0.4428	0.5054
NeoDTI [dim=128]	0.1981	0.1051	0.1961	0.4054
KGE_RF	0.2761	0.1859	0.3081	0.4679

TABLE 3.4: Comparison results of involved methods in the ablation study based on the DTINet dataset.

Part	Methods dim=16	Hits@10	Hits@3	Hits@1	MRR
Part1	FLONE_{base}	0.0596	0.0201	0.0080	0.0313
Part1	$\text{FLONE}_{ECFP-SEQ}$	**0.4367**	**0.3352**	**0.2379**	**0.3092**
Part2	FLONE_{base}	0.0000	0.0000	0.0000	0.0030
Part2	$\text{FLONE}_{ECFP-SEQ}$	**0.1872**	**0.1222**	**0.0779**	**0.1167**
Part3	FLONE_{base}	0.6473	0.5589	0.4635	0.5291
Part3	$\text{FLONE}_{ECFP-SEQ}$	**0.7054**	**0.5978**	**0.5111**	**0.5758**
Part	Methods dim=128	Hits@10	Hits@3	Hits@1	MRR
Part1	FLONE_{base}	0.1088	0.0363	0.0140	0.0494
Part1	$\text{FLONE}_{ECFP-SEQ}$	**0.5587**	**0.4528**	**0.3415**	**0.4171**
Part2	FLONE_{base}	0.0000	0.0000	0.0000	0.0028
Part2	$\text{FLONE}_{ECFP-SEQ}$	**0.2016**	**0.1561**	**0.1146**	**0.1483**
Part3	FLONE_{base}	0.6890	0.6163	0.5522	0.6000
Part3	$\text{FLONE}_{ECFP-SEQ}$	**0.7650**	**0.6870**	**0.6260**	**0.6723**

[4]

and reported the average evaluation metrics on each part separately. Specifically, the first part (Part 1) includes the samples corresponding to drugs that are in the original test set but not in the training set, and the second part (Part 2) includes the samples corresponding to targets that are in the original test set but not in the training set. The third part (Part 3) includes the samples corresponding to already seen drugs and targets. The numbers of unseen drugs, unseen targets, and samples for each independent repeat are in appendix A.3.

The results are provided in table 3.4, we observed that $\text{FLONE}_{ECFP-SEQ}$ clearly out-performed FLONE_{base} on all three parts. This indicated that introducing similarity-based drug and target encoders into FLONE not only improves its predictions on triplets related to seen drugs and targets, but also makes FLONE have the capability to provide effective predictions on ones corresponding to previously unseen drugs and targets. In addition, we further demonstrated the advantages of explicitly considering disease types when inferring targets of the given drug, through an extra experiment detailed in appendix A.6.

[4]The bold data indicates the best result under current evaluation metric and data.

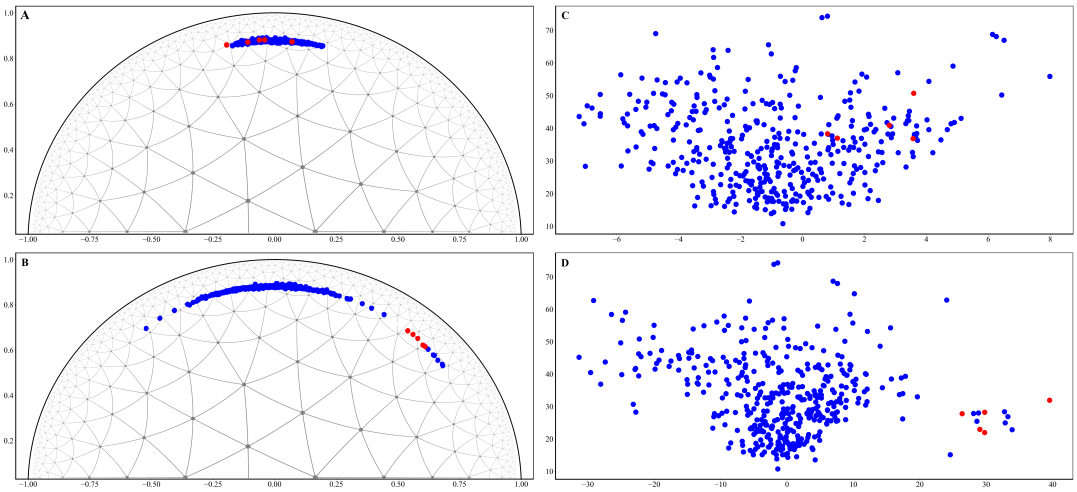


FIGURE 3.3: Visualization of the 2D embedding spatial layout of 128-dimension FLONE$_{ECFP-SEQ}$ and its Euclidean counterpart. These four layouts display all candidate target embeddings (under the given drug entity *Nitrazepam* and disease relation *leukemia*, *myeloid*, *acute*) in different situations (A)-(D). The blue points represent all candidate target embeddings, and the red points represent the embeddings of current positive targets in the test set under the given drug and disease (after dimension reduction). (A) The hyperbolic target embedding layout before applying the translation of the given disease in 2D Poincaré disk. (B) The hyperbolic target embedding layout after applying the translation of the given disease in 2D Poincaré disk. (C) The Euclidean target embedding layout before applying the translation of the given disease in 2D Euclidean space. (D) The Euclidean target embedding layout after applying the translation of the given disease in 2D Euclidean space. Besides, the black arc in layouts (A) and (B) is the boundary of 2D Poincaré disk.

3.3.6 Visualization of the embedding spatial layout

To give some insights for showing the captured information by hyperbolic embeddings, we investigated the difference of embedding spatial layout between the Lorentz KGC model and its Euclidean counterpart. Based on the 128-dimension FLONE$_{ECFP-SEQ}$ and FEC-FLONE$_{ECFP-SEQ}$ (on DTINet), we projected the Lorentz embeddings and Euclidean embeddings of all candidate target entities (i.e., the targets in the target entity set), for the given drug entity *Nitrazepam* (*DB01595*) and disease relation *leukemia*, *myeloid*, *acute*, to 2-dimension (2D) for visualization (figure 3.3). Specifically, the original Lorentz target embeddings were finally mapped into a 2D Poincaré disk in which the hyperbolic embedding quality can be effectively checked (Balazevic, Allen, and Hospedales, 2019), and the Euclidean ones were mapped to their 2D Euclidean space (the projection details are in appendix A.7).

For the 2D Poincaré disk, as shown in figure 3.3 (A) and (B), we draw the triangular tiling within it, these triangles had the same size in Euclidean space with

the vertices that can be treated as different feature embedding points. We can observe that, the space closer to the boundary of the Poincaré disk can include more embedding points, loosely speaking, there is more capacity to contain more embeddings points and to distinguish them (for downstream tasks) (Balazevic, Allen, and Hospedales, 2019). Through capturing the hierarchical structural information in the DDT network by learning our defined triplets, we found that the hyperbolic embeddings of the above targets (from FLONE$_{ECFP-SEQ}$) were successfully pushed close to the boundary of the 2D Poincaré disk, and after applying the translation offset of *leukemia*, *myeloid*, *acute*, every positive target embedding was clearly separated with all negative target embeddings. For the 2D spatial layout of the corresponding Euclidean target embeddings generated from the Euclidean counterpart of our Lorentz model (figure 3.3 (C) and (D)), we found that the Euclidean positive target embeddings were not as clearly separated from negative ones as those hyperbolic embeddings. This could indicate that the hyperbolic embedding is able to better capture the DDT triplets carrying hierarchical structural information in the DDT networks.

3.4 Chapter Summary

This chapter aims to demonstrate the effectiveness of introducing hyperbolic space to learn the drug-disease-target network topology with the consideration of additional disease types as well as drug and target priors for final target identification and screening. We have designed a fully hyperbolic Lorentzian-based GRL framework FLONE for the demonstration. The results on our curated datasets show that, not only the overall hyperbolic space introduction but also the detailed designs, including the explicit consideration of disease types based on triplet learning and the import of drug and target priors based on corresponding hyperbolic feature encoders, are beneficial to the accuracy of drug target predictions, which validates our hypothesis.

Chapter 4

Muthene: Improving therapeutic synergy score predictions with adverse effects using multi-task heterogeneous network embedding

This chapter aims to answer our second research question (given in section 1.2.2), and focuses on predicting regressive therapeutic drug-drug synergy scores based on capturing drug-target interaction mechanisms of actions (MoAs) (e.g., how a drug affects/interacts with its target to produce a therapeutic effect) with the auxiliary adverse effect learning. Section 4.1 provides the rationale behind our approach. Section 4.3 introduces our algorithm. Section 4.4 validates our approach on our curated dataset specifically designed for the formulated task. Finally, section 4.5 summarizes the chapter.

The main contents (including figures) of this chapter have been published in *Briefings in Bioinformatics* with the DOI: https://doi.org/10.1093/bib/bbac564 (Yue et al., 2023b).

4.1 Introduction

Drug combination is a more effective treatment than monotherapy for complex diseases such as cancers (Liu and Xie, 2021; Jia et al., 2009). Specifically, the use of multiple drugs that target different molecular mechanisms in the same cells not only

improves the overall therapeutic effect (TE) (Csermely et al., 2013; Jiang et al., 2020), but also reduces the required concentration of each drug which ultimately reduces the potential toxicity (Sun, Sanderson, and Zheng, 2016). However, drug combinations also could cause unexpected adverse effects (AEs) such as heart failure (Jiang et al., 2020; Zhang et al., 2013). Therefore, the elucidation of both the TEs and AEs of drug combinations is critical.

In recent years, many machine learning methods, especially those based on deep learning have been developed to select drug combinations with TEs or to predict drug pairs with AEs (Chen et al., 2016b; Wang et al., 2021). For example, the Deep Neural Network (DNN) based methods DeepDDI (Ryu, Kim, and Lee, 2018) and DeepSynergy (Preuer et al., 2018) were proposed to predict polypharmacy AEs and TEs, respectively. These methods can effectively reduce corresponding wet experiment costs by predicting high-confidence drug combinations (Liu and Xie, 2021).

However, most of these methods only considered TE or AE information separately, which might not be optimal. Can we combine TE and AE information to obtain better prediction results for TEs and AEs? We suspect the answer is positive. The first reason is that, superficially, both tasks use the drug chemical information and drug-target interaction information as the inputs (Ryu, Kim, and Lee, 2018; Preuer et al., 2018; Zitnik, Agrawal, and Leskovec, 2018), which indicates the relatedness of the tasks. By exploiting the relatedness between the tasks, we hypothesize that we might improve their learning efficiency and prediction accuracy. More profoundly, TEs and AEs are both measurable physiological changes to drug combination treatments, hence by considering TEs and AEs together, it is more likely to reveal their underlying MoAs, e.g., their common targets and the downstream molecular networks (Zhang et al., 2013), which could be beneficial to TE and AE prediction tasks.

In this study, we limit our scope to the TE prediction and propose the following hypothesis: TE prediction could benefit from the proper use of AE information. We first formulate the TE prediction as a multi-task drug (therapeutic) synergy score

regression problem (Zhang, Shen, Initiative, et al., 2012) that uses both AE and drug-target interaction (DTI) information. Since drug-target interactions are usually modelled as a heterogeneous complex network and are also heterogeneous to AE information, we propose Muthene, a multi-task heterogeneous network learning method specifically for synergy score predictions.

The main characteristic of the proposed Muthene algorithm is its ability to capture the MoA information for both TE and AE learning. Specifically, Muthene explicitly extracts four types of meta-paths as MoAs that represent the putative interaction pathways between the involved drugs and their shared targets. To learn the extracted meta-paths, Muthene uses a heterogeneous graph neural network (GNN)-based meta-path learning module with independent Bi-directional Gated Recurrent Unit (BiGRU) aggregators (Jiao, Sun, and Sun, 2018; Purkayastha et al., 2019). For the main task, i.e., the TE prediction, Muthene first uses this meta-path learning module to extract relevant drug-target interaction information, and then trains a regressor to predict synergy scores using drug combination cell line data. Simultaneously, Muthene also conducts the AE prediction as an auxiliary task using a similar procedure as the TE prediction task, except that the learning objective is AE classification instead of synergy score regression. In essence, this auxiliary task optimizes the model parameters for better synergy score predictions through backpropagation (Peng et al., 2022).

We conducted extensive experiments on Muthene. Our experimental results showed that the synergy score prediction benefited from the auxiliary AE prediction task, which is in line with our hypothesis. Apart from proposing and testing this novel hypothesis, our other contributions include:

- We collected and constructed a novel drug-drug interaction dataset with both TE and AE information for each sample to test our hypothesis.
- We also demonstrated that BiGRU could be a practical component for GNN to learn drug and target interaction pathway information, which might shed light on the MoAs of the drug combinations.

4.2 Materials

4.2.1 Datasets

Currently, there is no existing dataset to allow us to test the hypothesis that AE information is beneficial to synergy score predictions. To address this problem, we constructed a drug-drug interaction dataset that consisted of AEs, TEs, and their relevant information based on the following datasets: DrugComb dataset (mainly for TEs) (Zheng et al., 2021), TWOSIDES dataset (mainly for AEs) (Zitnik, Agrawal, and Leskovec, 2018), and Luo et al. dataset (for DTIs) (Luo et al., 2017). DrugComb is an online database that records the therapeutic effect degree of drug-drug pairs on cancer cell lines (in the form of drug-drug-cell line pairs), and the degree is measured based on four types of synergy scores separately, including Bliss (Bliss, 1939), Highest Single Agent (HSA) (Tan et al., 2012), Loewe (Loewe, 1953), and Zero Interaction Potency (ZIP) (Yadav et al., 2015). The original TWOSIDES dataset has 964 kinds of AEs occurring between different drug-drug pairs. Luo et al. dataset is a gold standard dataset in which 1,920 DTI pairs were selected.

Since DrugComb and TWOSIDES datasets contain TEs and AEs information respectively, we needed to generate a sub-set of drug-drug pairs that had both TEs and AEs information. In other words, every existing synergy score (based on a drug-drug-cell line pair) from DrugComb should have at least one known AE. To guarantee that each (drug-drug-cell line) sample has sufficient true labels, we only selected those corresponding to the top 20 frequent AEs and the top 60 frequent cell lines from the TWOSIDES and DrugComb separately.

We then integrated DTI pairs from DrugComb, TWOSIDES, and the Luo et al. dataset with the aforementioned sub-set, and ensured that every drug in these DTI pairs belonged to the drug set of the selected sub-set. In order to capture target-target interactions, e.g., shared pathways and cross-talk, we extracted and integrated the protein–protein interaction (PPI) network from TWOSIDES based on the targets in the selected DTI set. Next, to better capture the drug-drug interaction, we incorporated drug chemical structure information for every involved drug by generating Morgan extended connectivity fingerprints with a radius of 3 (ECFP6), which are circular topological fingerprints commonly used in drug repurposing tasks (Rogers

TABLE 4.1: The summary of our collected dataset.

Data Type	Total number
Synergy score types	4
Adverse effect types	20
Drug types	106
Cell line types	60
Gene types for gene expression data	677
Drug-drug-cell line synergy score sample number (4 types of synergy scores provided for each sample)	11,166
Drug-drug adverse effect sample number	2,446
Drug-target interaction number	2,332
Protein-protein interaction number	91,785

and Hahn, 2010; Landrum et al., 2013). In addition, we also integrated gene expression data to depict the biological variation in the selected 60 cell lines. Specifically, for every cell line, we retrieved expression values of genes (after 0-1 normalization) that encode the drug targets from the DepMap database (Tsherniak et al., 2017), and these targets belonged to the target set of the selected DTI set. A figure to illustrate the above steps is in figure B.1.

After the above steps with some data preprocessing, e.g., correcting naming errors and inconsistency, we obtained 11,166 drug-drug-cell line synergy scores (combinations) and 2,446 drug-drug AE samples, which shared the same 106 drugs. Meanwhile, we had the corresponding 2,332 DTIs, 91,785 PPIs, 106 drug ECFP6 features, and 677 gene expression values for each cell line, and the overall summary of our collected dataset is shown in table 4.1.

4.3 Methods

4.3.1 Construction of the heterogeneous therapeutic effect network

We can represent the TE-related drug-target relationships using a heterogeneous network denoted as $\mathbb{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $v_i \in \mathcal{V}$ denotes a node in this network, including drug (D) and protein (T); $\mathcal{E}$ is the set of edges (v_i, r, v_j), and $r \in \mathcal{R}$ is the type of edges that contains drug-drug TE relationships, DTIs, and PPIs. Specifically, the DTIs and PPIs are binary data, i.e., if there is an edge between the two nodes, the corresponding value is 1 (representing there is an effective edge), otherwise, it is 0.

TABLE 4.2: Description of biological meanings of four pre-defined meta-paths.

Meta-path names	Biological meanings
DTD	The start drug shares the same protein target with the end drug.
DTTD	The start drug and end drug interact with two protein targets, which also can interact with each other.
DTTTD	The start drug and end drug interact with two protein targets, which also can interact with another protein target.
DD (TE relationships)	The TE relationships between the start drug and end drug.

For drug-drug TE relationships, we converted the synergy score of each drug pair that was a real value to binary data using a threshold as detailed appendix B.2.

To explicitly capture the MoA, e.g., the drug-target and target-target interactions (PPIs) between a drug pair for both TE and AE learning, we used the concept of meta-path schema to mimic them, which is a specific combination pattern of edges from the start (source) drug node v_1 to the end (target or central) drug node v_{l+1}, denoted as $v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} ... \xrightarrow{r_l} v_{l+1}$ or $v_1 v_2 ... v_{l+1}$. Moreover, a meta-path instance is a sequence of nodes arranged following certain kind of meta-path schemas, in which the start and end drug nodes are connected by the edges in this instance, and they are called as a pair of meta-path neighbors.

Specifically, we considered four meta-path schemas to represent (TE-related) drug-drug interaction pathways in total, including: DTD, DTTD, DTTTD, and DD (TE relationships). The specific biological meanings of these schemas are shown in table 4.2.

Because the number of meta-path instances obtained after traversing the heterogeneous network grows exponentially with the length of the meta-path schema, to reduce the time and space complexity, we designed a heuristic sampling strategy. Specifically, we imposed an extra restriction to the 2nd protein (T) of DTTTD: this protein also needs to be a target of involved drugs in the network, and 50% uniform sampling ratio was used to choose the meta-path instances after applying this restriction. This could limit the meta-path instance choices for DTTTD into a more accurate and smaller range.

4.3.2 Overview of Muthene

Muthene consists of three modules: TE-related meta-path learning, AE prediction, and TE prediction modules. The illustration of Muthene is shown in figure 4.1.

First, Muthene constructs a heterogeneous TE network $\mathbb{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ as detailed in section 4.3.1. Then, for each drug-drug (-cell line) pair to be predicted in the network, Muthene generates all four types of meta-path instances for each drug node, to model MoAs for TE and AE learning. Second, in order to learn these meta-path instances, they are fed into the TE-related meta-path learning module to generate the meta-path embeddings of these drug nodes, which extract the interaction pathway information of the drug nodes. Third, Muthene generates the ECFP6 fingerprint for each drug node as its chemical feature embedding and then concatenates it to the drug meta-path embedding to create a drug integrated embedding. Fourth, the drug integrated embeddings and expression data of the cell line are fed to AE and TE prediction modules to predict the AE probability scores and therapeutic synergy scores of corresponding drug-drug-cell line pairs. We detailed each module below.

4.3.3 TE-related meta-path learning module

Acquisition of meta-path instance sets

For the sake of efficiently extracting the MoA related interactive characteristics from selected meta-path instances, based on the work of (Fu et al., 2020), our TE-related meta-path learning module was designed, which can consider the information of every drug and target node in the meta-path.

To generate a meta-path node embedding for each drug node (we need to generate an embedding under each type of pre-defined meta-paths for a drug node separately for its final meta-path embedding), we first had to obtain its meta-path instance sets as the input. Take the drug node D2 as an example, we traversed the whole heterogeneous TE network to acquire its four drug-related meta-path instance sets, i.e., DTD, DTTD, DTTTD, and DD sets. In each set, every instance is a sequence that starts from an arbitrary (start) drug node and ends at drug node D2 (D2 is denoted as the end/target/central node of this instance), following the node type arrangement of a specific meta-path schema. These instances in a set can construct

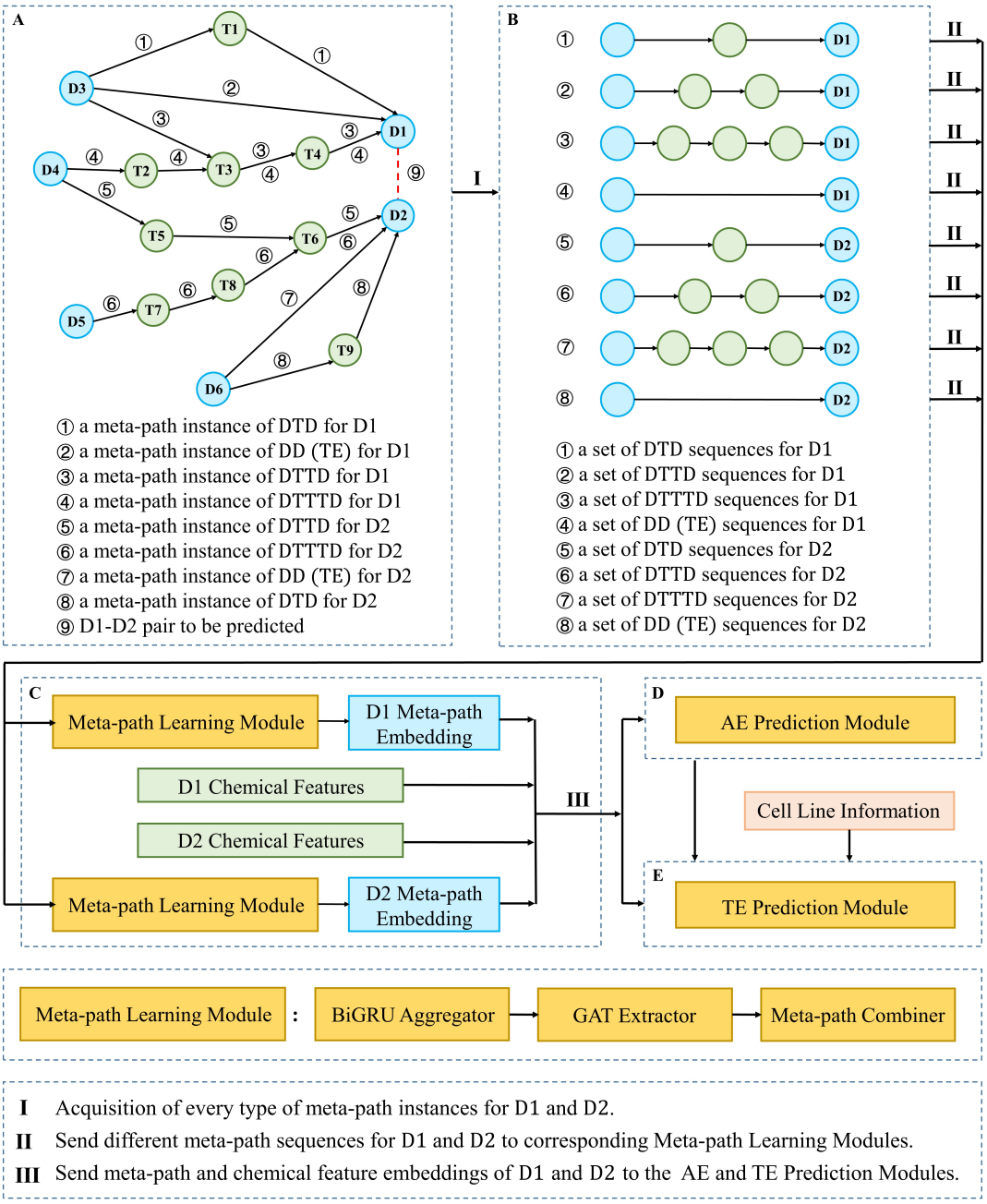


FIGURE 4.1: Illustration of Muthene. Take an example of predicting the synergy score between D1 and D2 in a certain cell line. the steps include: (A) Construct a heterogeneous TE network includes drug (D) and protein (T) nodes, and corresponding edges. (B) Acquire four types of meta-path instance sets for D1 and D2. (C) Send meta-path instance sets of D1 and D2 into the TE-related meta-path learning module respectively, to obtain D1 and D2 meta-path embeddings. Meanwhile, retrieve ECFP6 for D1 and D2. (D) Put all this information into the AE prediction module, to predict AEs between D1 and D2. (E) Put the input and output of the AE prediction module and expression data of the cell line into the TE prediction module, to calculate the final synergy score.

a sub-graph centered on D2 which captures the neighborhood information of D2 under a specific meta-path schema.

After obtaining four meta-path instance sets of a drug node, e.g., D2, we needed to give every node in the heterogeneous TE network a feature as the initial representation in GNN, thus, after this operation, every meta-path instance was transformed into a tensor. Specifically, the representations of all drug and target nodes in the heterogeneous TE network were initialized with the node-type-specific one-hot vectors to identify every node under its node category, and these vectors were filled into every meta-path instance according to the corresponding node type and node index. To ensure that the feature dimensions of all nodes in a meta-path instance were the same for the follow-up computation, we applied the node-type-specific transformation, for a node i with node type $t \in \mathcal{T}$, the transformation formula is as follows:

$$h_i^{t'} = W^t \cdot h_i^t \tag{4.1}$$

where h_i^t is the one-hot vector of this node, W^t is the trainable transformation matrix for the node type $t \in \mathcal{T}$, and $h_i^{t'}$ is the aligned vector of node i. Because the output dimension of every transformation matrix is set to be the same, thus for D2, after the process, D2 will have four 3D tensors corresponding to the four original meta-path instance sets, each one contains the interactive neighborhood information of a type of meta-path schema of D2 (for each tensor, the first dimension is the number of instances in the set, the second is the node number of the instance, the third is the output dimension); and this rule can be generalized to other drug nodes.

Generation of drug meta-path embeddings

The BiGRU aggregator The generated meta-path instance sets of each drug node will be processed by the corresponding meta-path-specific aggregator, which generates a new representation for each meta-path instance and its end node or central node (under one type of meta-path). The generated representations from the aggregators will be used to create the meta-path-specific embeddings for the corresponding end or central node in the next step. For simplicity, a meta-path instance starting

from node i, ending at node j, under the meta-path type $m \in M$ is denoted as $I_{m(i,j)}$.

We chose the BiGRU as the aggregator, intuitively, every $I_{m(i,j)}$ as a node feature sequence is reversible, for its end or central node j to be predicted, $I_{m(i,j)}$ and $I_{m(j,i)}$ contain different directions of interactive pathway information for it. In other words, it is semantically meaningful to learn $I_{m(i,j)}$ from the start node to the end node (i.e., extraction of the forward interaction information) and learn it from the end to the start (i.e., extraction of the backward interaction information), and BiGRU could capture such bi-directional information due to its nature (Jiao, Sun, and Sun, 2018). More specifically, we chose the final hidden state for $I_{m(i,j)}$ from BiGRU as the effective representation of $I_{m(i,j)}$, because it included all captured information from $I_{m(i,j)}$ on both forward and backward directions (denoted as $h^f_{m(i,j)}$). In addition, the updated representation of the end or central node of $I_{m(i,j)}$ can be obtained from the output of BiGRU (denoted as $h^l_{m(i,j)}$).

The GAT extractor To create meta-path-specific embeddings for corresponding end or central node, we adopted Graph Attention Networks (GAT), a spatial GNN (Kipf and Welling, 2016) proposed by (Veličković et al., 2017). The reason to choose GAT is that it can allocate more accurate weight for every meta-path-based neighbor's feature in the generation of the central node's embedding.

Similar to other spatial GNNs, GAT aggregates the neighbor nodes' features of each central node to generate the central node's low-dimensional representation. In our GAT, the central node is chosen as the end node of every $I_{m(i,j)}$, the neighbor used for the central node feature aggregation is the start node (i.e., the meta-path-based neighbor of the end node) in $I_{m(i,j)}$, and the final hidden state for $I_{m(i,j)}$ (i.e., $h^f_{m(i,j)}$) from BiGRU is treated as the feature of the used neighbor. To illustrate this, take the generation of D2 embedding under the meta-path DTD as an example: D2 aggregates its all meta-path-based neighbors' features in the DTD-based subgraph, in which the feature of D2 itself is obtained from BiGRU and the meta-path-based neighbors' features are replaced by the corresponding meta-path instances' representations (as these representations contain more comprehensive information than the original neighbors' features).

More specifically, our GAT method generates embeddings of central nodes for each meta-path schema independently. For a central node j under meta-path $m \in M$, the formula for calculating its embedding h_j^m is as follows:

$$\alpha_{ij}^m = \frac{exp(\sigma(a_m^T \cdot [h_{m(i,j)}^l \| h_{m(i,j)}^f]))}{\sum_{k \in N_j^m} exp(\sigma(a_m^T \cdot [h_{m(k,j)}^l \| h_{m(k,j)}^f]))} \tag{4.2}$$

where α_{ij}^m is the allocated weight for node j's meta-path-based neighbor i in the feature aggregation process (for generating node j's embedding under meta-path m). σ represents the activation function, a_m^T is a trainable attention vector for meta-path m, N_j^m represents all meta-path-based neighbors of node j under meta-path m.

After acquiring every α_{ij}^m for node j, we ran a weighted aggregation to generate the embedding of node j under meta-path m (i.e., h_j^m). To make the GAT extractors more stable, we executed the weighted aggregation K times independently, and concatenated the output embedding from each weighted aggregation to create the new embedding, where $\|$ denotes the concatenation operation:

$$h_j^m = \|_{k=1}^K \sigma(\sum_{i \in N_j^m} \alpha_{ij}^m \cdot h_{m(i,j)}^f) \tag{4.3}$$

The outputs from GAT extractors are meta-path-specific embeddings. Take D2 as an example, after running GAT, we could obtain four (DTD, DTTD, DTTTD, DD (TE relationships)) meta-path-specific embeddings for D2, these embeddings will be fused to acquire the final meta-path embedding for D2 in the next step.

The attention mechanism for integrating meta-path-specific embeddings After obtaining the meta-path-specific embeddings, for each drug central node, we needed to combine its embeddings based on the four meta-path types into one embedding that captured composite interaction information. To handle the heterogeneous characteristics of different meta-paths, we adopted an attention mechanism (Fu et al., 2020; Wang et al., 2019b) (termed as the meta-path combiner) to weighed-sum the embeddings under different meta-paths automatically.

The attention mechanism starts by calculating the importance weight w_t^m for fusing meta-paths of node type $t \in \mathcal{T}$ as follows:

$$w_t^m = \frac{1}{|V_t|} \sum_{i \in V_t} q_t^T \cdot tanh(W_t \cdot h_i^m + b_t) \tag{4.4}$$

where V_t is the node set of type t, q_t^T is the trainable attention vector for node type t, W_t is the meta-path combination transformation matrix for node type t, h_i^m is the embedding of node i (with current node type) under meta-path m defined in formula equation (4.3), and b_t is the trainable bias vector for node type t.

Next, we normalized the importance weight of every meta-path $m \in M$ using softmax:

$$\beta_t^m = \frac{exp(w_t^m)}{\sum_{p \in M}^{M} exp(w_t^p)} \tag{4.5}$$

Utilize the normalized weight, the summarized embedding of node i with type $t \in \mathcal{T}$ can be calculated as follows:

$$h_i^t = \sum_{m \in M}^{M} \beta_t^m \cdot h_i^m \tag{4.6}$$

Finally, a non-linearity is added to h_i^t, for generating the final meta-path embedding z_i^t, where W_P^t is the projection matrix of node type t. Based on this, the meta-path embedding of every involved drug can be obtained.

$$z_i^t = \sigma(W_P^t \cdot h_i^t) \tag{4.7}$$

4.3.4 Integration of drug chemical information

Drug chemical features (e.g., ECFP6) are also important for drug-drug combination related predictions (Huang et al., 2020). However, it is inappropriate to use them as the node initialization of the TE-related meta-path learning module. The reason

is that, the heterogeneous TE network includes drug and target nodes, using drug chemical features in the case that the target nodes cannot be represented as such features will be harmful to the unity of the initial node embedding space (Yu et al., 2021a). Thus, we used ECFP6 of each drug node i as its independent chemical feature embedding (denoted as z_i^C), which was concatenated with the corresponding meta-path embedding to produce the integrated embedding of this drug.

4.3.5 The adverse effect probability score prediction module

To predict the probability of each of 20-selected AEs between a drug-drug pair, the AE prediction module combines the integrated embedding of each drug in a drug-drug pair to calculate the AE probability score:

$$P_{ij}^{AE} = \sigma(W_{AE}[z_i^{drug}, z_i^C, z_j^{drug}, z_j^C]) \tag{4.8}$$

where W_{AE} is a parameterized decoder. P_{ij}^{AE} is a 20-dimensional vector, with each element representing the occurrence probability of a certain AE. Besides, this is a multi-label classification task, thus the binary cross entropy (BCE) loss is adopted to evaluate the difference between the predictions and corresponding ground truths, where x_n is the predicted AE probability score for the n_{th} sample, y_n is the corresponding AE label ground truth, and N denotes the sample number.

$$l_{BCE} = mean(\{-[y_n \cdot logx_n + (1 - y_n) \cdot log(1 - x_n)]\}, n \in 1, ..., N) \tag{4.9}$$

4.3.6 The therapeutic synergy score prediction module

In our compiled dataset, the synergy scores between drug-drug pairs are recorded based on the selected 60 cell lines, thus the gene expression data for each cell line is also added into computation to mimic the biological variation in different cell lines, for effectively distinguishing them (Talloen et al., 2007).

Specifically, for obtaining cell line k's gene expression input z_k^{cell}, a fully connected layer is used to reduce its original 677-dimension to a fixed dimension d^{cell}. The synergy score of the drug i-drug j-cell line k pair is calculated as follows:

$$P_{ijk}^{TE} = DNN_{TE}[z_i^{drug}, z_i^C, z_j^{drug}, z_j^C, z_k^{cell}, P_{ij}^{AE}] \tag{4.10}$$

in which the corresponding AE prediction is explicitly added as the DNN_{TE} input. DNN_{TE} is the DNN having the same basic structure as the DeepSynergy by (Preuer et al., 2018), for decoding the complex drug-, cell line-, and AE-related input (i.e., Muthene uses the same basic structure as DeepSynergy to decode this information). Specifically, DeepSynergy was constructed based on the conic layers (where each layer had a half number of neurons compared with the last layer), and rectified linear unit (ReLU) activation (Nair and Hinton, 2010) was additionally applied to every intermediate layer. Furthermore, to avoid the gradients were propagated from the TE prediction task back to AE prediction task in the training phase, we treated the passed P_{ij}^{AE} as a constant.

To evaluate the model prediction error for this task, we selected the loss function mean square error (MSE) as follows, in which y_n represents the synergy score ground truth of the n_{th} sample, x_n is the corresponding predicted value from the model, and N is the sample number.

$$l_{mse} = mean(\{(x_n - y_n)^2\}, n \in 1, ..., N) \tag{4.11}$$

To summarize, we optimized the whole framework including the three main modules in an end-to-end fashion, which made the framework better share all the effective information at once. To facilitate the training process, due to the different scale of l_{BCE} and l_{MSE} (i.e., each drug-drug-cell line sample will generate a BCE loss and a MSE loss for guiding the optimization, and these two losses have different value scales), we used the hyper-parameters α to control the weight between the two losses, for combining them as the overall optimization objective function:

$$l_{total} = \alpha l_{MSE} + (1 - \alpha) l_{BCE} \tag{4.12}$$

It is worth mentioning that, Muthene should not distinguish drug *i*-drug *j*-cell line *k* and drug *j*-drug *i*-cell line *k* pairs, as they have the same biological meaning. Thus, we generated each sample twice (with the different drug-drug pair order) in the training set, and the estimated values in validation and test sets were the arithmetic average of predictions of the corresponding two samples.

4.3.7 Model evaluation settings

To evaluate our method, it is critical to avoid the information data leakage problem (Kaufman et al., 2012), which generates over-optimistic but misleading results. However, the existing model evaluation settings, i.e., randomly splitting (drug-drug-cell line) samples into training, validation, and test sets suffer from this problem. To illustrate this problem, suppose we have two drug-drug-cell line pairs $D_i - D_j - C_{K1}$ and $D_i - D_j - C_{K2}$, if they are allocated into training and test sets separately, after training, when predicting the synergy score of $D_i - D_j - C_{K2}$, the model "has already seen" the information about the $D_i - D_j$ combination in the training phase.

To avoid this pitfall, we split drug-drug-cell line pairs based on drug-drug pairs. In other words, the drug-drug-cell line pairs with the same drug-drug pair were put into the same set (i.e., training, validation, or test sets). In this case, drug-drug pairs or combinations in the test set did not occur in the training set, which mitigated the information leakage problem.

Based on this setting, drug-drug-cell line pairs corresponding to 6:2:2 of all drug-drug pair varieties were allocated into training, validation, and test sets, respectively, in which the training set was used for building models, the validation set was used for optimizing parameter settings, and the test set was used for testing models and producing corresponding performance evaluation results. This procedure was repeated five times independently, for each time, before splitting data, our whole dataset was randomly shuffled to make different drug-drug pair varieties enter each

set. We computed and reported the average evaluation metrics over the five independent repeats.

In addition, each drug-drug-cell line pair had 20 AE labels and a synergy score value. In our study, we conducted our experiments based on the collected four types of synergy scores (i.e., Loewe, Bliss, HSA, and ZIP) respectively. As the synergy score prediction is a regression task, we used MSE (mean square error), MAE (mean absolute error), and the Pearson's correlation coefficient (abbreviated as Pearson) as metrics to evaluate the model performance on the test set.

4.4 Results

4.4.1 AE prediction task benefits synergy score predictions

The main objective of our experiments is to test the hypothesis on whether the AE information benefits synergy score predictions in our multi-task heterogeneous network learning framework. To this end, we removed the AE prediction module from Muthene to create a variant named Muthene-AE. For further investigation, we included three representative methods DeepDDS (Wang et al., 2022b), TranSynergy (Liu and Xie, 2021), and DeepSynergy (Preuer et al., 2018). To illustrate the difference between Muthene and these methods, we listed them in table 4.3. To examine whether AE information would benefit the single-task learning methods such as DeepSynergy, we included AE binary labels of drug pairs as an extra part of its model input, and termed this variant as DeepSynergyAE. The model running environment and hyper-parameter settings are provided in appendix B.3, and the experimental results are shown in table 4.4. In addition, all involved methods were run under the same random seed.

From the results, we found that, based on MSE, MAE, and Pearson, Muthene achieved overall better performance compared with Muthene-AE no matter on Loewe, Bliss, HSA, or ZIP scores. The above clearly verified the effectiveness of our proposed multi-task framework and corresponding basic hypothesis. It is interesting to see that, the results of DeepSynergyAE were not always better than those of DeepSynergy among every type of synergy scores. This indicated that, at least

[1]The bold data indicates the best result on current evaluation metric and synergy score type.

TABLE 4.3: Description of the involved comparison methods.

Method Name	Related Description
DeepDDS	DeepDDS uses GNNs and multi-layer perception to encode the drug molecular graph and cell line expression data to generate drug and cell line embeddings or features respectively. Multiple fully connected layers are then used to process the concatenation of the generated drug and cell line embeddings for therapeutic effect predictions. Here following their settings, we used DeepChem (Ramsundar et al., 2019) tool to generate the initial input graphs of GNNs for our collected drugs.
TranSynergy	TranSynergy takes drug target interaction information (as drug features) and cell line features (e.g., gene expression) as the model input, and utilizes dimension reduction layers followed by a Transformer (Vaswani, 2017) to encode them. The synergy scores are also predicted by fully connected layers.
DeepSynergy	DeepSynergy is detailed in (section 4.3.6), which is a DNN-based method. In our experiments, it took the concatenation that contains ECFP6 (as drug features) of drug pairs and the cell line gene expression as the model input (the used ECFP6 is same to that used in Muthene).

the simple use of the AE information (e.g., feature concatenation) is not beneficial enough to the single-task learning method DeepSynergy.

Furthermore, we observed that among the four types of synergy scores for quantifying therapeutic effects, Loewe consistently achieved a higher Pearson regression coefficient, which indicated that it is more suitable to be a learning objective for these machine learning methods, thus, we adopted Loewe to conduct the following experiments. We also discussed the performance of the mentioned three representative methods in (appendix B.4).

4.4.2 The effectiveness of selected network components

Based on standard Muthene with the same basic hyper-parameters (shown in appendix B.3) and experimental settings, as well as Loewe synergy score, we did the ablation study for the important added components in this multi-task framework. First, to demonstrate the efficiency of using BiGRU as the aggregator for GNN to extract drug and target interactive pathway information from pre-defined meta-paths

TABLE 4.4: The averaging evaluation results of involved methods over the five independent repeats.

Synergy Score: Loewe	MSE	MAE	Pearson
Muthene	**180.6258**	**8.9157**	**0.6353**
Muthene-AE	189.2802	9.0448	0.6201
DeepSynergy (Preuer et al., 2018)	203.9883	9.6107	0.5678
DeepSynergyAE (Preuer et al., 2018)	208.3976	9.6589	0.5741
DeepDDS (Wang et al., 2022b)	201.2541	9.6429	0.5714
TranSynergy (Liu and Xie, 2021)	273.3114	11.5534	0.3130
Synergy Score: Bliss	MSE	MAE	Pearson
Muthene	**45.7406**	**4.3330**	**0.5503**
Muthene-AE	48.4344	4.3810	0.5155
DeepSynergy (Preuer et al., 2018)	49.6929	4.4324	0.5052
DeepSynergyAE (Preuer et al., 2018)	49.1170	4.4236	0.5093
DeepDDS (Wang et al., 2022b)	61.2581	4.7372	0.3225
TranSynergy (Liu and Xie, 2021)	57.7050	4.5757	0.3595
Synergy Score: HSA	MSE	MAE	Pearson
Muthene	**30.2391**	**3.6613**	**0.4672**
Muthene-AE	31.2363	3.7435	0.4343
DeepSynergy (Preuer et al., 2018)	31.9054	3.7471	0.4111
DeepSynergyAE (Preuer et al., 2018)	31.2234	3.7430	0.4301
DeepDDS (Wang et al., 2022b)	35.7116	3.9316	0.3121
TranSynergy (Liu and Xie, 2021)	35.2235	3.8679	0.2925
Synergy Score: ZIP	MSE	MAE	Pearson
Muthene	**29.2428**	**3.6736**	**0.5225**
Muthene-AE	29.6737	3.6852	0.5142
DeepSynergy (Preuer et al., 2018)	30.1284	3.7027	0.5061
DeepSynergyAE (Preuer et al., 2018)	30.5356	3.7137	0.4936
DeepDDS (Wang et al., 2022b)	34.7283	3.9232	0.4010
TranSynergy (Liu and Xie, 2021)	34.1866	3.8451	0.3890

TABLE 4.5: Ablation study results based on involved Muthene variants.

Methods	MSE	MAE	Pearson
Muthene	**180.6258**	**8.9157**	**0.6353**
Mut-GRU	191.5294	9.3234	0.6002
Mut-Mean	186.3361	9.0604	0.6279
Mut-GIN	195.8967	9.3091	0.6019

bi-directionally, we replaced BiGRU with Gated Recurrent Unit (GRU, i.e., unidirectional aggregator) (Cho, 2014) and mean function (i.e., non-directional aggregator) to create two variants called Mut-GRU and Mut-MEAN. Specifically, for Mut-GRU, all $h^l_{m(i,j)}$ and $h^f_{m(i,j)}$ (in equation (4.2) and equation (4.3)) were generated from GRU. For Mut-MEAN, $h^l_{m(i,j)}$ and $h^f_{m(i,j)}$ were obtained from the original central node's feature described in section 4.3.3 and were obtained by averaging the features of all nodes in the corresponding meta-path instance, respectively.

We also implemented another variant, denoted as Mut-GIN, to test the effectiveness of using learnt chemical feature embeddings of drugs to replace original ECFP6 features in our method. Specifically, we used a powerful GNN named Graph Isomorphism Network (GIN) (Xu et al., 2018) to learn the molecular graph of each drug, for generating its new chemical feature embedding. The relevant detailed description can be found in appendix B.5, and the GIN module was trained along with other modules in an end-to-end way. The evaluation results of these variants are shown in table 4.5.

We observed that the standard Muthene better-performed Mut-GRU and Mut-MEAN, suggesting that BiGRU did learn more useful drug and target interactive pathway information from the meta-paths. Meanwhile, the performance of Mut-GIN was inferior to that of standard Muthene, which indicated that using the drug chemical feature embedding produced from molecular graphs (by GNNs) may not be necessary in our task. In addition, we did an extra ablation study about the performance influence caused by various feature dimensions of meta-path embeddings (generated based on the heterogeneous TE network), which is shown in appendix B.6.

[2]The bold data indicates the best result on current evaluation metric.

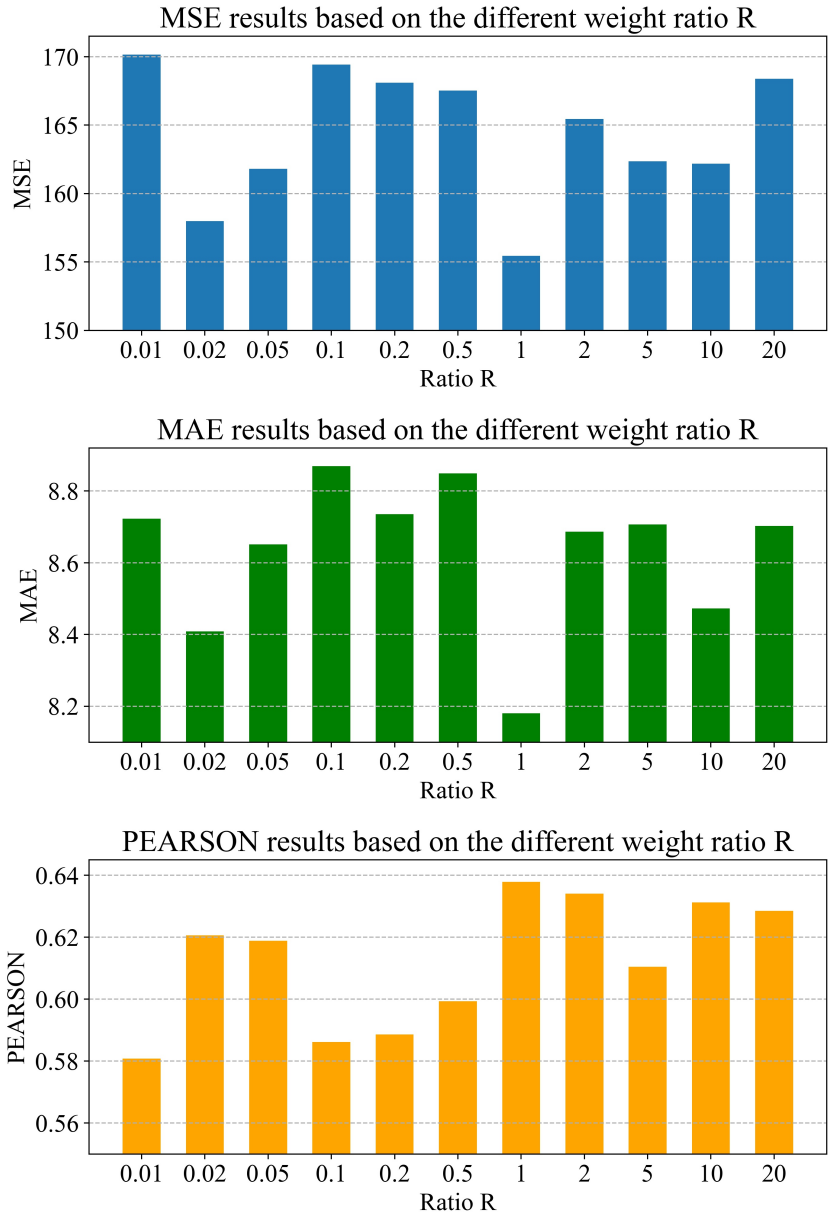


FIGURE 4.2: Different model evaluation results given various weight ratio R.

4.4.3 The effects of the weights between the two losses

There is one important hyper-parameter α which is used to weighted combine the BCE loss and MSE loss of each sample for better optimizing Muthene. We, therefore, based on Loewe synergy score, evaluated the sensitivity of α, through adjusting α while fixing the random seed and other hyper-parameters during one of the five times data splitting. The model evaluation results under different weight ratio R (i.e., $\frac{\alpha}{1-\alpha}$) are shown in figure 4.2.

Based on the results, we can observe that, α was relatively sensitive, with the

increase of ratio R, Muthene gradually achieved overall better Pearson's regression coefficient until R reached at a threshold, and then it started to get worse. This indicated that, within a proper R scale, increasing the weight ratio R could benefit the improvement of synergy score prediction accuracy. In addition, experimentally, we found that the search range of R = [0.01, 0.05, 0.1, 0.5, 1, 5, 10] is suitable to find the value of R to demonstrate our hypothesis.

4.4.4 MoAs Elucidation: A Case Study

To demonstrate how the defined meta-paths reveal MoAs for TE predictions. We used a drug-drug pair in the test set, *Vincristine—Dasatinib* pair, which shows a significant (top ranked) drug synergy effect on Melanoma cell lines *A2058* and *UACC62*.

Since Muthene generated the meta-path embedding for each drug separately by aggregating meta-path instances of the drug, we collected the intersection of the meta-path instances between these two drugs. Not surprisingly, there is no DTD instance, i.e., they have different types of drug targets: *Vincristine* binds to tubulin proteins to stop the tubulins from forming microtubules, and *Dasatinib* inhibits several tyrosine kinases. However, these two drugs shared the same 114 DTTD instances and 340 DTTTD instances. From the 114 DTTD instances, we identified the most frequent meta-path pattern: *Vincristine* links to a tubulin protein, e.g., TUBB3, and then links to a kinase family RIPK (Receptor-interacting serine/threonine-protein kinase), e.g., RIPK1, finally links to *Dasatinib*. It is interesting to know that while both the tubulin protein and RIPK kinase families are the key regulators in cell death, their MoAs are different: Tubulin proteins cause apoptosis (Zimprich et al., 2004) while RIPK kinases regulate necroptosis (Liu et al., 2011). This finding suggested that by modulating cell death through different MoAs, e.g., apoptosis and necroptosis, the *Vincristine—Dasatinib* pair provides a more effective treatment of advanced metastatic Melanoma and could overcome potential drug resistance (Agalliu et al., 2015).

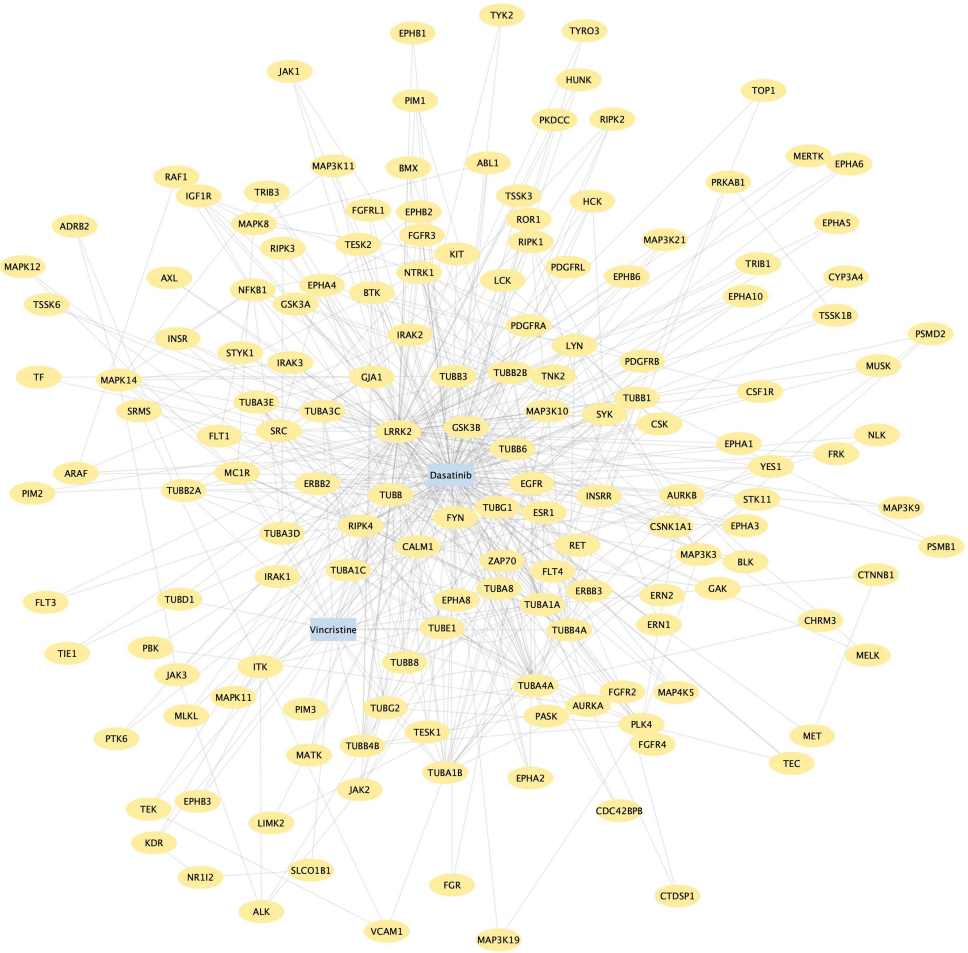


FIGURE 4.3: The network constituted by all DTTTD instances shared by *Vincristine* and *Dasatinib*.

By inspecting the 340 DTTTD instances, we found another interesting MoA hypothesis. Among all the targets of the DTTTD instances, Leucine-rich repeat kinase 2 (LRRK2), a target that bridges tubulin targets of *Vincristine* and kinase targets of *Dasatinib* (e.g., RIPK1 and RIPK4), appears more frequently than other (target) nodes. A Betweenness Centrality analysis of the network generated from these DTTTD instances (see figure 4.3) confirmed that LRRK2 has the highest Betweenness Centrality, which means that it is the central or hub node of the network. From a complex network perspective, a central node should play an important role in network functions, i.e., Melanoma progression in our case. However, LRRK2 is a well-known gene associated with Parkinson's disease (Zimprich et al., 2004). How does LRRK2 appear in the DTTTD instances as the putative MoAs for treating Melanoma? A meta-analysis showed that Parkinson's disease patients have a high risk of getting Melanoma (Liu et al., 2011), and a subsequent genomic study confirmed the mutation of LRRK2 is associated with Melanoma (Agalliu et al., 2015). Interestingly, our Muthene algorithm independently uncovered this association (through using the meta-path schema). More importantly, Muthene also hypothesized that the high synergy between *Vincristine—Dasatinib* pair could be explained by the indirect modulation of LRRK2 by *Vincristine*, which subsequently enhances the regulation of the kinase targets of *Dasatinib*.

4.5 Chapter Summary

This chapter aims to demonstrate the feasibility of explicitly modelling of drug-target interaction MoAs (using pre-defined meta-paths) combined with auxiliary adverse effect learning, for final screening of drug pairs with better synergistic therapeutic effects. To achieve this, we have collected the specialized dataset and further designed a multi-task meta-path-based heterogeneous GNN framework Muthene for the demonstration. It is observed that, both the proper design of meta-paths and their embedders and the additional adverse effect classification task contribute to the further performance improvement of drug-drug therapeutic synergy score predictions. This demonstrates the effectiveness of our hypothesis.

Chapter 5

MpbPPI: A multi-task pre-training-based equivariant approach for the prediction of the effect of amino acid mutations on protein-protein interactions

In this chapter, we start tapping from the large-scale 2D biological networks to the small-scale 3D macro-molecular structures to enable the geometric graph representation learning for finer screenings on more similar protein candidates (e.g., identifying the top dozen candidates produced by single-site or multiple-site amino acid mutations/substitutions). More specifically, we investigate how to design an effective yet flexible geometric learning framework for screening better protein mutants (binders) which could share stronger binding intensity to the aiming target. Section 5.1 introduces our hypotheses that serve as the motivation for our approach. Section 5.2 describes an overview of our proposed algorithm and provides the experimental results underlying the algorithm design. Next, section 5.3 and section 7.1.3 provide the further analysis and more implementation details and intuitions behind the algorithm. Finally, section 5.4 summarizes the chapter.

The main contents (including figures) of this chapter have been published in *Briefings in Bioinformatics* with the DOI: https://doi.org/10.1093/bib/bbad310 (Yue

et al., 2023c).

5.1 Introduction

Protein–protein interactions (PPIs) play a crucial role in many biological processes, including antibody-antigen binding, cell apoptosis, and signal transduction (Wang, Cang, and Wei, 2020; Braun and Gingras, 2012; Feng et al., 2017). Alterations in PPIs can affect the formation of multi-protein complexes, causing disruption to both intercellular communication networks and intracellular signaling pathways, ultimately contributing to the development of diseases such as cancer and drug resistance (Porta-Pardo et al., 2015). Amino acid (AA) mutations are a primary cause of PPI alterations, with different mutations leading to different disturbances in PPIs, and resulting in different phenotypic outcomes. Understanding the potential mechanisms by which AA mutations impact PPIs is vital for developing therapies that target these interactions (Barouch et al., 2013; Ben-Kasus et al., 2007; Hiew et al., 2023; Chen and Zhou, 2005). Although various experimental methods are available to identify the effects of mutations, they are often costly and time-consuming. To overcome this challenge, fast and reliable computational alternatives are necessary (Barderas et al., 2008).

The change in binding affinity caused by AA mutations, usually quantified as a scalar $\Delta\Delta G$, represents the difference in binding affinity between wild-type (WT, unmodified protein) and mutant-type (MT, protein that has undergone mutations) protein complexes (Jankauskaitė et al., 2019). Over the years, numerous computational approaches have been developed to predict $\Delta\Delta G$ (Xiong et al., 2017). Early $\Delta\Delta G$ tools mainly employed physical energies as features to describe PPIs, such as FoldX (Schymkowitz et al., 2005), Rosetta (Kortemme, Morozov, and Baker, 2003), and BeAtMuSiC (Dehouck et al., 2013). These methods rely on a (linear) combination of physical energies dominated by van der Waals interaction, hydrogen bond, electrostatic interaction, solvation, etc. terms. However, many of these methods faced several limitations, including insufficient conformation sampling in the mutation region and conformational efficiency (Liu et al., 2021).

In recent years, the rapid expansion of experimental data and computational resources led to the emergence of machine learning methods, especially those mainly focusing on characterizing geometric properties of protein structures (Li et al., 2016). The structure of protein dictates its biochemical function (Alberts et al., 2002), thus the combination of fully captured protein structural characteristics and the powerful non-linear fitting capability of machine learning methods could provide a more accurate approximation of PPI $\Delta\Delta G$ (Rodrigues et al., 2019). For instance, aiming to tackle $\Delta\Delta G$ predictions caused by AA single point mutations, (Geng et al., 2019) developed a random forest-based algorithm iSEE to process PPI interface structure features enhanced by evolutionary conservation and energy-based information. Based on the specially-designed graph-based structural signatures representing the residue environment, an online $\Delta\Delta G$ analysis webserver named mCSM-PPI2 was built by (Rodrigues et al., 2019). Furthermore, (Wang, Cang, and Wei, 2020), (Liu et al., 2023), and (Wee and Xia, 2022) proposed their algebraic topology-based methods, respectively, to produce the simplified physical structural representations of protein-protein complexes with the consideration of AA site mutation information. The generated representations were fed into different ensemble decoders to calculate the final $\Delta\Delta G$.

In addition to traditional machine learning methods, the deep learning-based geometric method graph neural network (GNN) could be a more efficient way for extracting structural representations of protein-protein complexes. This is because it can directly take the geometric graph which can be easily constructed from raw protein coordinates as model inputs (Jiang et al., 2023). A typical processing pipeline for such methods involves acquiring a protein-protein complex graph after AA site mutations, then inputting the WT and mutant protein-protein complex graphs into a GNN to obtain the respective graph structure representations. The $\Delta\Delta G$ is then predicted by comparing these two representations using an independent decoder. However, due to the difficulty in acquiring mutant complex structure ground truth, corresponding simulated structures or assumed proxies are usually needed. For example, (Liu et al., 2021) cropped the original atom-level WT and computationally-simulated mutant complex graphs based on a pre-defined distance threshold. They then input these cropped graphs into a pre-trained GNN encoder, to produce the

final low-dimensional representation that captures the structural difference between WT and mutant complexes. (Jiang et al., 2023) designed an end-to-end GNN framework named DGCddG which takes assumed WT and mutant graphs equipped with multiple biochemical features for $\Delta\Delta G$ predictions.

Most machine learning and deep learning methods currently rely on a combination of energy-based and structural-based features to predict the $\Delta\Delta G$ value associated with a given PPI mutation (Rodrigues et al., 2019). However, energy-based features can be computationally expensive, while many existing structural-based features focus on modelling single residue-wise contact relationships or inter-atom interactive relationships, which may not capture more complete geometric relationships between the backbone, side chains, and their interactions within AA residues. The relative positions and orientations of residues, as well as the interactions between their side chain and backbone atoms, can significantly impact the alignment of amino acids within a protein structure and influence protein-protein interactions.

Moreover, transfer learning, which aims to capture general regulations of downstream samples through pre-training the model on more accessible other data (not involving the type of labels in downstream tasks) (Hu et al., 2019; Dong et al., 2021; Devlin, 2018), can help overcome the challenge of limited experimental mutation datasets for $\Delta\Delta G$ prediction. It is crucial to ensure that the pre-training dataset is carefully selected and prevent unintended information leakage to downstream tasks, which could result in over-optimistic model evaluation results. Meanwhile, when predicting PPI $\Delta\Delta G$ using machine learning or deep learning algorithms, it is also important to consider the impact of amino acid mutations on protein structure (Liu et al., 2021). Mutant structures are able to provide valuable insights into how the amino acid substitution affects the protein structure, which can serve as input to these algorithms to predict the $\Delta\Delta G$. Many methods are available for generating the mutant structures, including empirical models [FoldX (Schymkowitz et al., 2005)], homology modeling [MODELLER (Webb and Sali, 2016)], SwissModel [Schwede et al., 2003)], and deep learning methods [AlphaFold2 (Jumper et al., 2021)], among others. Each method has its strengths and limitations, and the choice of method can affect the accuracy of the mutant structure and, subsequently, the $\Delta\Delta G$ prediction. Therefore, it is crucial to carefully evaluate and compare different methods and

choose the most appropriate method for a given PPI system.

To summarize, the more efficient and sufficient characterization and learning of comprehensive geometric relationships within PPI systems (in which the mutant structures are unknown), under insufficient labelled PPI $\Delta\Delta G$ data is needed. We solved these issues by introducing a flexible geometric equivariant graph neural network framework MpbPPI that can accommodate various mutant sources, for the precise prediction of PPI $\Delta\Delta G$ resulting from single and multiple missense mutations. Specifically, to capture more comprehensive PPI geometric information, we designed two types of residue-level contact graphs that include the different-scale geometric interactive relationships between residues, between side chain atoms, and between side chain atoms and residue backbone atoms. To better utilize unlabeled PPI structures to produce low-dimensional protein-protein complex representations with good generalization for limited number of downstream samples, we first proposed a geometric equivariant encoder (GEE) contained in MpbPPI to encode the above contact graphs, and it will learn effective PPI geometric regulations through our specially-designed multi-task pre-training technique performed on a strictly screened pre-training dataset. Furthermore, MpbPPI serves as a framework compatible with multiple mutant generation tools, enabling flexible handling of mutant structures including ones led by multiple missense mutations.

We conducted extensive experiments on MpbPPI, to evaluate its performance in conjunction with three distinct mutation structure generation tools: FoldX, MODELLER, and AlphaFold2. Our approach achieved promising performance on three single mutation datasets and one more challenging multiple mutations dataset under two realistic evaluation settings. The results demonstrate that MpbPPI is a promising approach for predicting protein-protein binding affinity alterations caused by AA mutations.

5.2 Results

5.2.1 Overview of MpbPPI

As an overview (figure 5.1), MpbPPI fully learns the geometric regulations of protein-protein complexes based on the pre-training dataset (comprised of protein complexes sharing both a low topological similarity and sequence identity to the complexes in the downstream datasets, details on the data collection can be found in appendix C.1), by first generating two types of residue-level contact graphs that characterize different inter-residue geometric interactive relationships. Specifically, for each protein-protein complex in our strictly screened pre-training dataset, MpbPPI creates both a residue-level *K*-nearest neighbor (KNN) and radius contact graphs that contain both residue backbone and side chain atom-related geometric information (see section 5.3 for details). The graph structure of two contact graphs is used to model the relatively large-scale residue-level geometric relationships, while the residue backbone and side chain atom geometric information within it (as residue graph node features) is used to sufficiently capture the relatively small-scale complex interactive relationships between various atoms in a single residue.

Next, to better facilitate the learning of geometric regulations of the PPI structures, we devised four pre-training tasks (to be performed simultaneously) for the protein geometric equivariant encoder (GEE) of MpbPPI, including (residue) backbone denoising, side chain denoising, solvent-accessible surface area (SASA) prediction, and AA type prediction. Intuitively, multiple pre-training objectives aiming at learning geometric properties of individual residues themselves from different perspectives will encourage the GEE to capture the comprehensive interactive relationships within a complex, which is ultimately beneficial to generate high-quality PPI structural representations for downstream analyses. Based on this intuition, we corrupted the contact graphs of each pre-training protein-protein complex by adding noise to the 3-dimensional (3D) coordinates of the randomly chosen residue backbone and side chain atoms, and the corrupted graphs were sent to the GEE to perform the multi-task pre-training.

After the pre-training, MpbPPI first generates the aforementioned contact graphs for the WT and corresponding mutant complexes (which are from various mutant

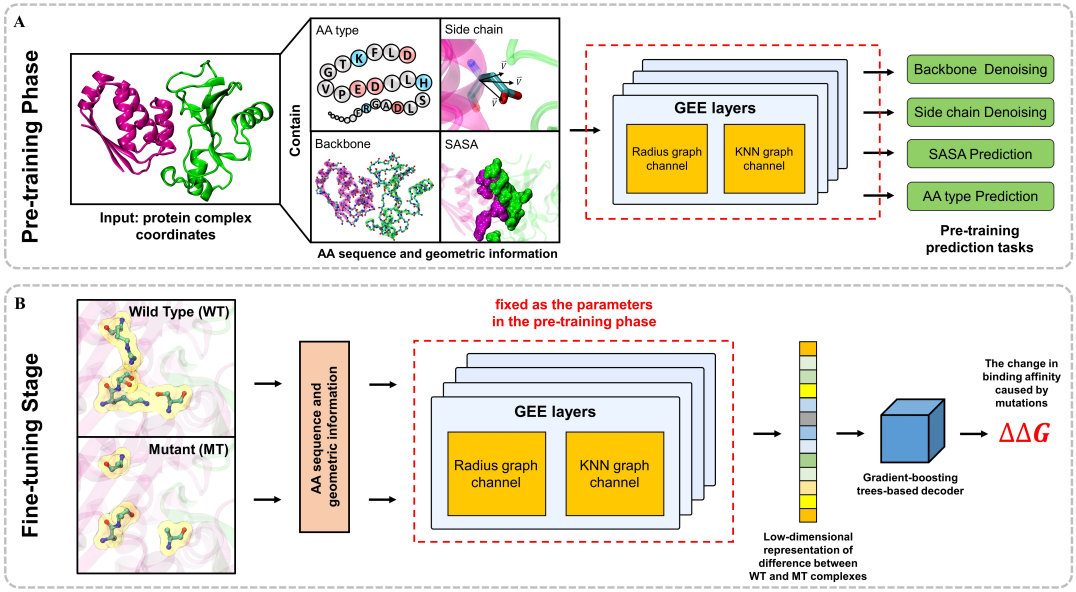


FIGURE 5.1: The flowchart of the MpbPPI framework. The framework consists of a pre-training phase and a downstream prediction phase. Specifically, for each pre-training and downstream sample point, MpbPPI generates the residue-level KNN and radius contact graphs, which contain different-scale residue backbone and side chain geometric information of the corresponding protein-protein complex structure (see section 5.3 for details). In the pre-training phase [Panel (A)], the proposed GEE encoder learns the geometric regulations of protein-protein complexes through our defined four geometric property-related denoising (i.e., recovery) tasks. In the downstream prediction phase [Panel (B)], MpbPPI uses a GBT-based decoder to predict PPI $\Delta\Delta G$ for a WT-MT complex pair based on their encoded representations from the pre-trained GEE encoder.

generation tools) of each sample point in the downstream $\Delta\Delta G$ dataset. Based on employing message passing to propagate different-scale geometric information within the contact graphs (figure 5.2), GEE encoder produces the low-dimensional structural representation for comparing the geometric difference between WT and mutant complexes. Finally, MpbPPI employs the gradient-boosting trees (GBT) which is an ensemble predictor specialized at handling overfitting, to predict the PPI $\Delta\Delta G$ for a WT-MT complex pair from the generated representation.

The overall diagram further illustrating the data flow of MpbPPI and how its components are linked is shown in figure 5.2, and more details and explanations of the whole MpbPPI framework can be found in section 5.3.

5.2.2 MpbPPI yields accurate PPI $\Delta\Delta G$ prediction

To comprehensively compare MpbPPI with other existing advanced approaches, we considered three single AA site mutation benchmark datasets, including single-point mutations in the AB-Bind dataset (Pires and Ascher, 2016) (S645), non-redundant interface single-point mutations in the SKEMPI dataset [11] (Xiong et al., 2017), and single-point mutations in the SKEMPI2 dataset (Rodrigues et al., 2019) (S4169). We also incorporated one more challenging benchmark dataset that contains multiple-point AA mutations (as multiple-point mutations could bring more complex PPI system conformation changes compared with single-point mutations): complete sample points of the AB-Bind dataset (Sirin et al., 2016), denoted as M1101.

Mutation-level cross-validation results

Based on the above four datasets, following the convention, we first performed the tenfold cross-validation splitting data based on mutation sample points, and used mean Pearson's correlation coefficient (abbreviated as R_P here) as the main evaluation metric (Wang, Cang, and Wei, 2020). In other words, all mutation sample points in each dataset will be uniformly split into 10 folds for the tenfold cross-validation. Additionally, for each dataset, we ran such tenfold cross-validation five times independently. For each repeat, the whole dataset was randomly shuffled to make different mutation sample points enter each fold, and we reported the average results of these independent runs. Under this evaluation setting, in order to examine

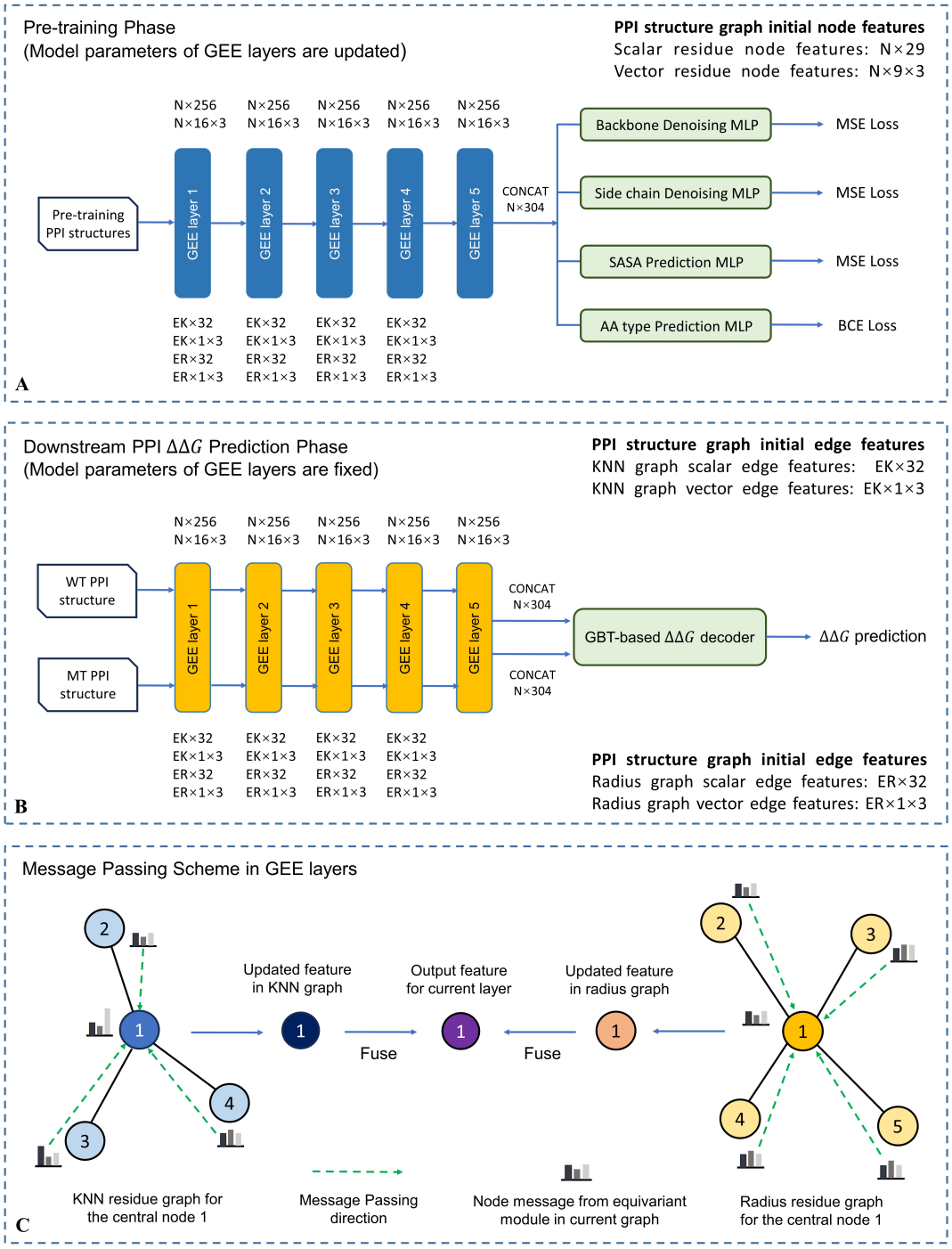


FIGURE 5.2: Panel (A) illustrates the MpbPPI data flow in the pre-training phase. In this phase, the pre-training protein-protein complex (residue number: N) represented by the KNN (edge number: EK) and radius contact graphs (edge number: ER) is sent to a five-layer GEE encoder. Based on message propagation, the encoder outputs the updated embedding of every residue node in the current complex, which will be simultaneously sent to four multi-layer perceptions (MLPs) specific to different pre-training tasks to guide the model optimization. The input and output dimensions of each intermediate layer are shown around this layer. For downstream ΔΔG prediction phase [Panel (B)], WT and mutant PPI structures represented by the same type of contact graphs as above are sent to the trained GEE to produce separate residue node embedding sets, which are then sent to the GBT-based decoder to predict the final ΔΔG for current sample point (see section 5.3). Panel (C) illustrates the basic message propagation scheme in each GEE layer, in which the similar operations will be performed to each (central or target) residue node in the protein-protein complex.

TABLE 5.1: Comparison of the R_P of the baseline methods on downstream PPI $\Delta\Delta G$ datasets. The results of GeoPPI (Liu et al., 2021), TopNetTree (Wang, Cang, and Wei, 2020), DGCddG (Jiang et al., 2023), PerSpect-EL (Wee and Xia, 2022) are the tenfold cross-validation results reported in their original papers.

S645	R_P	M1101	R_P
MpbPPI-FoldX	0.615±0.013	MpbPPI-FoldX	**0.787±0.002**
MpbPPI-MODELLER	0.598±0.006	MpbPPI-MODELLER	0.780±0.001
MpbPPI-AlphaFold2	0.577±0.015	MpbPPI-AlphaFold2	0.765±0.005
DGCddG	-	DGCddG	-
PerSpect-EL	0.590	PerSpect-EL	-
GeoPPI	**0.650**	GeoPPI	0.780
TopNetTree	**0.650**	TopNetTree	-
FoldX	0.271	FoldX	0.273
Flex ddG	0.067	Flex ddG	0.059
MM/GBSA	0.039	MM/GBSA	0.018
S1131	R_P	S4169	R_P
MpbPPI-FoldX	**0.865±0.003**	MpbPPI-FoldX	**0.795±0.004**
MpbPPI-MODELLER	0.847±0.002	MpbPPI-MODELLER	0.782±0.005
DGCddG	0.848	DGCddG	-
PerSpect-EL	0.853	PerSpect-EL	-
GeoPPI	0.850	GeoPPI	0.780
TopNetTree	0.850	TopNetTree	0.790
FoldX	0.423	FoldX	0.326
Flex ddG	0.217	Flex ddG	0.136
MM/GBSA	0.303	MM/GBSA	0.164

1

the performance of MpbPPI based on different mutant structure generation tools, we created three variants of MpbPPI with different suffixes indicating the mutant complex source (i.e., MpbPPI-FoldX, MpbPPI-MODELLER, MpbPPI-AlphaFold2, the detailed description of mutant complex generation is in appendix C.2), and compared them with seven baseline methods for PPI $\Delta\Delta G$ predictions, including GeoPPI (Liu et al., 2021), TopNetTree (Wang, Cang, and Wei, 2020), DGCddG (Jiang et al., 2023), PerSpect-EL (Wee and Xia, 2022), FoldX (Schymkowitz et al., 2005), Rosetta macromolecular modelling suite (Flex ddG) (Barlow et al., 2018), and MM/GBSA (Genheden and Ryde, 2015; Wang et al., 2019a; Chen et al., 2016a). The former four methods are representative advanced geometric-based machine learning methods, while the latter three are mainstream energy-based methods. The detailed description of these methods can be found in appendix C.3.

Our experimental results are shown in table 5.1, in which the results of GeoPPI (Liu et al., 2021), TopNetTree (Wang, Cang, and Wei, 2020), DGCddG (Jiang et al., 2023), and PerSpect-EL (Wee and Xia, 2022) were obtained from the original papers. We also reported more experimental results, including the root mean square error (RMSE) and mean absolute error (MAE) of our methods and additional analysis, in table C.1, figure C.1, and figure C.2. In summary, based on the five times of independent runs of the tenfold cross-validation, MpbPPI (MpbPPI-FoldX) achieved overall better performance compared with other competitive methods, while only on the S645 dataset, the second-best result was achieved. Another important advantage of MpbPPI is that it can tackle multiple-point mutation cases more easily because of its flexibly direct input of the mutant structure (graph) from various sources.

In addition, for MpbPPI (MpbPPI-FoldX), when compared with GeoPPI, another geometric property pre-training-based method, under the strict pre-training sample screening and the same GBT predictor (see section 5.3), MpbPPI still overall outperformed it on the relatively larger single- and multiple-mutation datasets. This further demonstrated the effectiveness of the proposed different-scale geometric encoding framework and corresponding pre-training tasks. Interestingly, MpbPPI-FoldX consistently outperformed MpbPPI-MODELLER and MpbPPI-AlphaFold2, which indicated that under current PPI $\Delta\Delta G$ prediction benchmark datasets, FoldX could be a more appropriate tool as the source of mutant PPI protein-protein structures (the further analysis about these generation tools are provided in section 7.1.3). Furthermore, we did extra experiments about 1) directly correlating the difference between sequence representations of the mutant and wildtype complexes to the PPI $\Delta\Delta G$ value, 2) reproducing results of some representative baseline methods based on available information. These contents are provided in appendix C.6 and appendix C.7, respectively.

[1]The bold data indicates the best result under current evaluation metric and dataset. Due to the high computational cost of AlphaFold2 (the computational cost analysis of the mutant generation tools is provided in section 7.1.3), we tested performance of MpbPPI-AlphaFold2 based on the relatively smaller S645 and M1101 datasets.

The effectiveness of devised geometric characteristics and their pre-training tasks

Based on the best-performing model MpbPPI-FoldX, under the same experimental and evaluation setting as section 5.2.2, we conducted the ablation study to discuss the effectiveness of our devised geometric characteristics and corresponding pre-training tasks. Specifically, we compared different variants of MpbPPI-FoldX, including MpbPPI (Backb+AA), MpbPPI (Backb+Sidec+AA), MpbPPI (Backb+SASA+AA), and MpbPPI (Backb+Sidec+SASA+AA), in which the suffixes represent different combinations of the used geometric characteristics plus their pre-training techniques (in other words, each suffix represents the presence of the corresponding residue node feature and pre-training task). Besides, to demonstrate the effectiveness of the auxiliary AA prediction task, we removed this prediction task from MpbPPI (Backb+Sidec+SASA+AA), and also put this variant into the comparison (denoted as MpbPPI (w/o AA)). The evaluation results of these variants on each dataset are shown in table 5.2 (we also reported the RMSE and MAE results in table C.3).

From the results we found that, compared with MpbPPI (Backb+AA) that did not use any explicit characteristics depicting geometric relationships between side chain atoms and between side chain and residue backbone atoms, the inclusion of side chain denoising and SASA prediction pre-training tasks (and corresponding features, see section 5.3), i.e., MpbPPI (Backb+Sidec+AA) and MpbPPI (Backb+SASA+AA) improved the performance. This clearly demonstrated the importance of our designed two types of side chain-related geometric learning pre-training tasks (and corresponding features). Besides, the performance margin between MpbPPI (w/o AA) and MpbPPI (Backb+Sidec+SASA+AA) illustrates the effectiveness of the auxiliary AA prediction task. Furthermore, as for the best performing combination under the current experimental setting, it could vary for different datasets, e.g., the combination of Backb+SASA+AA achieved the best result on S4169, but for S645, M1101, and S1131, the best combinations were Backb+Sidec+SASA+AA, Backb+Sidec+SASA+AA, Backb+Sidec+AA, respectively.

[2]The bold data indicates the best result under current evaluation metric and dataset.

TABLE 5.2: Comparison of the R_P of five MpbPPI-FoldX variants on each downstream dataset. The results of MpbPPI-FoldX variants were reported based on the average of five times of the independent tenfold cross-validation executions.

S645	R_P
MpbPPI (Backb+Sidec+SASA+AA)	**0.615±0.013**
MpbPPI (Backb+SASA+AA)	0.575±0.016
MpbPPI (Backb+Sidec+AA)	0.549±0.008
MpbPPI (Backb+AA)	0.507±0.016
MpbPPI (w/o AA)	0.613±0.013
M1101	R_P
MpbPPI (Backb+Sidec+SASA+AA)	**0.787±0.002**
MpbPPI (Backb+SASA+AA)	0.761±0.007
MpbPPI (Backb+Sidec+AA)	0.755±0.005
MpbPPI (Backb+AA)	0.738±0.003
MpbPPI (w/o AA)	0.785±0.002
S1131	R_P
MpbPPI (Backb+Sidec+SASA+AA)	0.854±0.002
MpbPPI (Backb+SASA+AA)	**0.865±0.003**
MpbPPI (Backb+Sidec+AA)	0.848±0.003
MpbPPI (Backb+AA)	0.838±0.003
MpbPPI (w/o AA)	0.831±0.003
S4169	R_P
MpbPPI (Backb+Sidec+SASA+AA)	0.793±0.003
MpbPPI (Backb+SASA+AA)	**0.795±0.004**
MpbPPI (Backb+Sidec+AA)	0.775±0.003
MpbPPI (Backb+AA)	0.766±0.004
MpbPPI (w/o AA)	0.766±0.003

[2]

The prediction results in the more challenging test scenario

In this section, we further tested the generalization capability of MpbPPI through a WT protein-protein complex type-based data splitting setting. Specifically, to better examine the model performance on previously unseen WT protein-protein complexes (the training and test sample points in the above mutation-level cross-validation may share relatively more WT protein-protein complex types), we devised a data splitting method in which original sample points were split into five folds and there was no intersection of WT protein-protein complex types between any of the two folds. We strived to make the both sample total number and WT protein-protein complex type number assigned to each fold to be as close as possible. Furthermore, we performed fivefold cross-validation based on this data splitting scheme, in this case, the model cannot see any WT protein-protein complexes (types) in the test set during the training phase, which enables model performance examination in the more realistic application scenario. We performed such data splitting five times, for each repeat, making different WT protein-protein complex types enter each fold, and we reported the averaging results of these five times of independent (fivefold) cross-validation runs. A detailed description of the above data splitting can be found in appendix C.9. For performance comparison, we considered available GeoPPI, FoldX, Flex ddG, and MM/GBSA. For GeoPPI, for a fair comparison, we used its original released pre-trained encoder and GBT hyper-parameters on S4169 to reproduce the results on the S4169 dataset based on the same data splitting. In addition, to investigate the better MpbPPI setting in this challenging scenario, we included both MpbPPI (Backb+Sidec+SASA+AA) and MpbPPI (Backb+SASA+AA), since they achieved the best results on different downstream datasets in section 5.2.2.

Our comparative study showed that (see figure 5.3 in which MpbPPI (Backb+Sidec+SASA+AA) and MpbPPI (Backb+SASA+AA) were abbreviated as MpbPPI_BSSA and MpbPPI_BSA separately. The RMSE and MAE results were reported in table C.4), MpbPPI exhibited more accurate predictive capability when facing previously unseen WT protein-protein complexes, which can further demonstrate the effectiveness of our method. Furthermore, we observed that

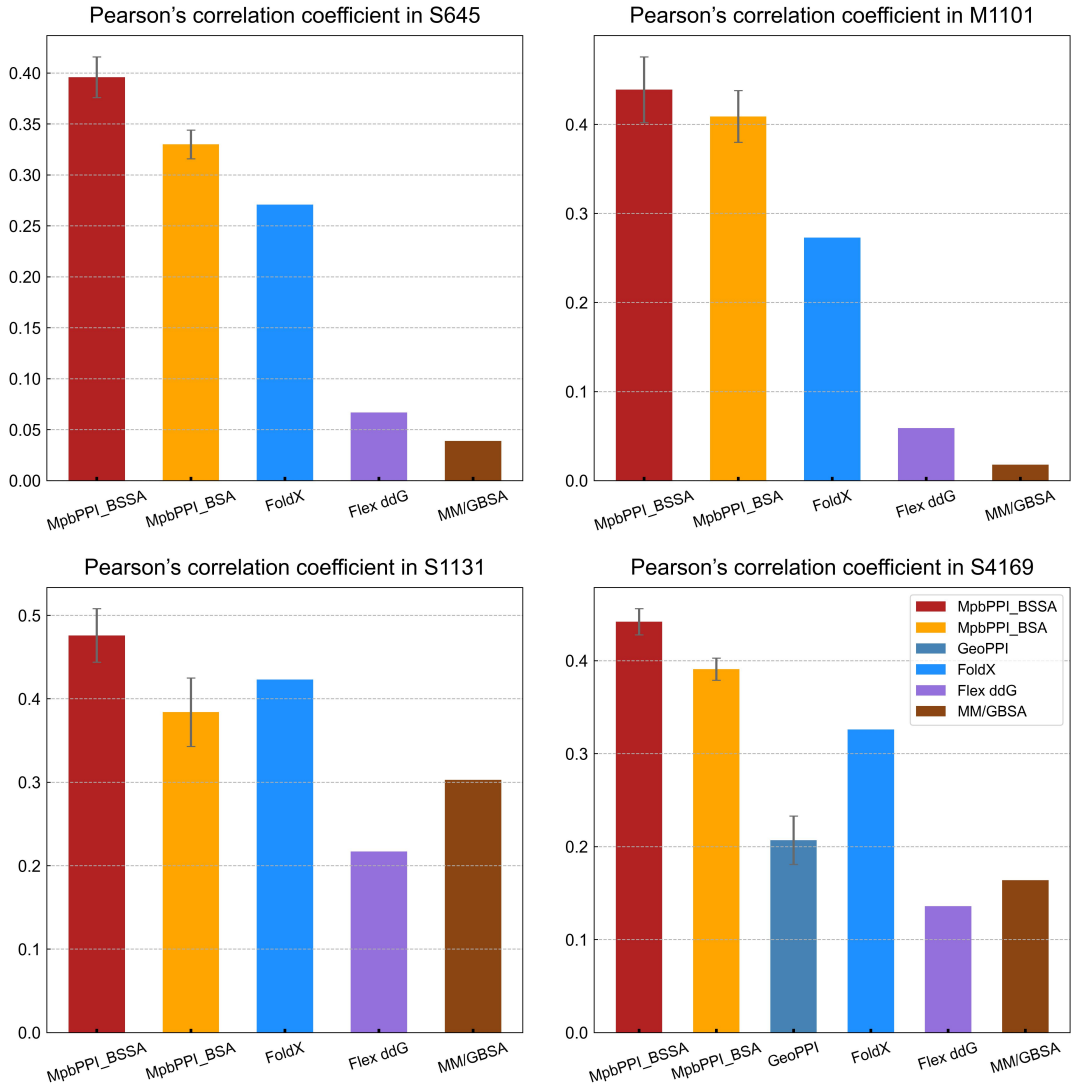


FIGURE 5.3: MpbPPI outperformed other involved methods for PPI $\Delta\Delta G$ prediction under the five-time WT protein-protein complex-based cross-validations. We reported the experimental results on each dataset based on the main evaluation metrics R_P. For the machine learning-based methods, the results were expressed as mean±SD, while for the empirical energy-based methods, the results were expressed as the mean value. MpbPPI (Backb+Sidec+SASA+AA) and MpbPPI (Backb+SASA+AA) were abbreviated as MpbPPI_BSSA and MpbPPI_BSA.

MpbPPI_BSSA outperformed MpbPPI_BSA on all downstream datasets (the performance margins were 20%, 7.3%, 24%, and 13% on S645, M1101, S1131, and S4169 for R_P, respectively), indicating that the integration of a combination of the two devised side chain-related geometric characteristics can bring better model generalization in current task setting. Based on this, we recommend adopting MpbPPI (Backb+Sidec+SASA+AA) in an actual application setting.

5.3 Methods

5.3.1 Generation of the refined residue-level contact graphs for protein-protein complexes

Before the generation of the residue-level contact graphs, to ensure that each residue node contains complete atom geometry coordinate information, we used FoldX to complete all side chains (based on raw PDB files) in both the pre-training and downstream sets. At the same time, all H atoms were removed from these PDB files.

We then modelled a given PPI complex as a residue contact graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of residues in the PPI complex (the position of each residue node is determined by its $C\alpha$ 3D coordinate) and $\mathcal{E}$ is the set of edges between them. The procedure for defining $\mathcal{E}$ is outlined below:

In order to capture more comprehensive interactive relationships for residue nodes in $\mathbb{G}$ [i.e., considering different inter-residue geometric contact relationships with different edge distributions (Zhang et al., 2022)], we defined an independent K-nearest neighbor (KNN) graph $\mathbb{G}^K = (\mathcal{V}^K, \mathcal{E}^K)$ (where each node i connects other K nodes nearest to it, based on the Euclidean distance between node position coordinates) and a radius graph $\mathbb{G}^R = (\mathcal{V}^R, \mathcal{E}^R)$ (where each node i connects all other nodes within a R-radius 3D sphere centered at i) for each protein-protein complex. For each $\mathbb{G}^K$ and $\mathbb{G}^R$, K and R are set to 20 and 10Å, and every residue node i in both $\mathbb{G}^K$ and $\mathbb{G}^R$ are equipped with two coordinate sets C_i^B and C_i^S containing the backbone atom (N, $C\alpha$, C, O) coordinates and side chain atom coordinates (removing H), respectively (for a given residue, in the both KNN and radius graphs, the contained residue nodes and initial C_i^B and C_i^S are the same).

Graph node features

In order to generate (initial) residue node and edge features f_i and $f_{(i,j)}$ for $\mathbb{G}$, we first calculated the centroid of $\mathbb{G}$ based on its all raw backbone atom (N, $C\alpha$, C, O) coordinates, and preliminarily normalized all the backbone and side chain coordinate sets by subtracting the calculated centroid coordinate value (to get $C^{B'}$ and $C^{S'}$, representing all backbone and side chain atom sets in current $\mathbb{G}$). Next, in order to fully characterize the geometric relationships between side chain atoms and between side chain and residue backbone atoms within the residue i from different perspectives, node features f_i were defined by concatenating the following seven feature sets:

- The centroid of side chain atoms (vector)
- The center of mass of side chain atoms (vector)
- The maximum coordinate value of side chain atoms (vector)
- The unit vector of $C\alpha$-side chain geometric relationships (vector)
- The normalized $C\alpha$ coordinate (vector)
- Solvent-accessible surface area (SASA) (scalar)
- Residue interface information (scalar)

The detailed calculation description and meaning of these features are provided in appendix C.13.

Other than the above seven types of residue node features, we also added five additional node features for providing AA sequence information and close neighbouring relationships of a residue, in which $C\alpha_i^{'}$ and $C\beta_i^{'}$ represent the $C\alpha$ and $C\beta$ coordinate of residue i after the preliminary centroid-based normalization:

- The amino acid (AA) type, given as a one-hot representation (scalar)
- Chain index (scalar)
- Sine-cosine encoded dihedral angles (scalar)
- Unit vector encoding of the $C\alpha_i^{'}$ close neighbours (vector)
- Unit vector encoding of the $C\alpha_i^{'} - C\beta_i^{'}$ orientation (vector)

5.3.2 Graph edge features

To create edge features $f_{(i,j)}$ capturing interactive contact relationships between residue nodes in current $\mathbb{G}$ (here denoting edges from nodes j to i as (i, j)), we concatenated the following feature sets:

- The unit vector in the direction of $C\alpha_j^{'} - C\alpha_i^{'}$ (vector)
- Euclidean distance encoding between $C\alpha_j^{'} - C\alpha_i^{'}$ based on the Gaussian radial basis functions (scalar)
- Sine-cosine encoded $j - i$ node index (scalar).

For the sine-cosine and Gaussian radial basis encoding methods, we suggest referring (Jing et al., 2020) for the detailed description.

For the scalar and vector features of a residue node i, we concatenated them separately, to produce the node scalar features f_i^S ($R \in \mathcal{V} \times 29$), node vector features f_i^V ($R \in \mathcal{V} \times 9 \times 3$), edge scalar features $f_{(i,j)}^S$ ($R \in \mathcal{E} \times 32$, from residue node j to i), and edge vector features $f_{(i,j)}^V$ ($R \in \mathcal{E} \times 1 \times 3$, from residue node j to i), which will be encoded by our GEE layers (as the input) independently. Please note that due to the different generation principles of geometric interactive edges, the edge features of residue i in the KNN and radius contact graphs are different (while the initial node features are exactly the same in the two graphs).

5.3.3 The GEE encoder and its pre-training techniques

The equivariance property is of importance in protein-related representation learning based on machine learning. Specifically, for a protein structure, its learnt representation should not be sensitive to the imposition of protein's rigid motions (e.g., extra translation to original protein 3D coordinates), because the physical law controlling the dynamics of molecules is not influenced by such motions (Wu et al., 2022a; Ganea et al., 2021; Huang et al., 2022). In order to incorporate such a property into our framework flexibly, we considered employing an equivariant-based graph neural network (GNN) backbone GVP-GNN (Jing et al., 2020), which uses a message passing strategy (Gilmer et al., 2017 for geometric neighbor-based learning of

(residue) node-level representations. However, GVP-GNN cannot effectively process the heterogeneous graphs in which nodes or edges could have multiple types, and so unable to satisfy our requirement of performing the KNN and radius graph learning simultaneously (the original equations of GVP-GNN are provided in appendix C.14). To solve this problem, we extended GVP-GNN into a multi-relational mode as follows to create our geometric equivariant encoder (GEE) layer (in which the superscripts KNN and $radius$ represent the feature or message specific to corresponding contact graphs separately):

$$f_{(i,j)}^{KNN_message} = GVP(concat(f_j^{(S,V)}, f_{(i,j)}^{KNN(S,V)})) \tag{5.1}$$

$$f_{(i,j)}^{radius_message} = GVP(concat(f_j^{(S,V)}, f_{(i,j)}^{radius(S,V)})) \tag{5.2}$$

$$f_i^{KNN_aggregation} = mean(\sum_{j:\, edges\, of\, j\, to\, i \in \mathcal{E}\, in\, KNN\, graph} f_{(i,j)}^{KNN_message}) \tag{5.3}$$

$$f_i^{radius_aggregation} = mean(\sum_{j:\, edges\, of\, j\, to\, i \in \mathcal{E}\, in\, radius\, graph} f_{(i,j)}^{radius_message}) \tag{5.4}$$

$$f_i^{(S,V)} = LayerNorm(f_i^{(S,V)} + Dropout(concat(f_{(i,j)}^{KNN_aggregation}, f_{(i,j)}^{radius_aggregation}))) \tag{5.5}$$

$$f_i^{(S,V)} = Layernorm(f_i^{(S,V)} + Dropout(GVP(f_i^{(S,V)}))) \tag{5.6}$$

Here we adopted the notation $f_j^{(S,V)}$ and $f_{(i,j)}^{(S,V)}$ to refer to the tuples of f_j^S and f_j^V and the tuples of $f_{(i,j)}^S$ and $f_{(i,j)}^V$, respectively. GVP is a special neural network module which is mathematically demonstrated to be insensitive to 3D rigid motions (Jiang et al., 2020). $f_{(i,j)}^{message}$ represents the message representation flowing or aggregating from residue node j to i during message passing. Intuitively, the message passing of geometric characteristics with different scales in different PPI contact graphs could generate enough powerful representation for the downstream PPI $\Delta\Delta G$ predictions.

In addition, we stacked five GEE layers (i.e., equations (5.1) to (5.6)) to constitute the GEE encoder to increase its ability to non-linear fitting, and the hidden feature dimensions for f_j^S, f_j^V, $f_{(i,j)}^S$, $f_{(i,j)}^V$ in each layer were the same, i.e., 256, 16, 32, 1, respectively (figure 5.2 (A)-(B)).

Based on the above GEE encoder and the designed contact graphs for each protein-protein complex in our strictly-screened pre-training set, our multi-task pre-training strategy was devised, in which the backbone denoising, side chain denoising, SASA prediction, and AA type prediction tasks were included, which will be performed simultaneously. Intuitively, setting multiple pre-training tasks, which share closely-related geometric information of the residues themselves, could be beneficial to capturing comprehensive interactive contact relationships (regulations) within a protein-protein complex. Please see the detailed description and meaning of these tasks in appendix C.15. Concisely, for each protein-protein complex in the pretraining set, we first randomly selected 15% of its residues and set a (boolean) mask to these residues. For these masked residues, we added noise to their backbone and side chain atom coordinates to corrupt the complex's overall conformation, and then the corrupted conformation (information) was used to recover the complex's corresponding ground truth conformation or properties (based on the task-specific multi-layer perceptions).

Besides, according to the ground truth to be recovered, the loss functions measuring the model reconstruction errors were mean square error MSE (for backbone denoising, $MSE^{C\alpha}$), MSE (for side chain denoising, MSE^{Sidec}), MSE (for SASA prediction, MSE^{SASA}), and binary cross entropy BCE (for AA type prediction, BCE^{AA}), respectively. To summarize, we can perform the above four pre-training tasks simultaneously in an end-to-end fashion (figure 5.2 (A)). The overall learning objective for guiding the model optimization is formulated as follows, and the Adam (Kingma, 2014) with the initial learning rate of 0.001 is adopted as the corresponding optimizer:

$$l^{overall} = MSE^{C\alpha} + MSE^{Sidec} + MSE^{SASA} + BCE^{AA} \tag{5.7}$$

Furthermore, the choice of the masked residues and magnitude of noise (the details of noise injection is in appendix C.15) is selected "on-the-fly". In other words, for every epoch in the pre-training, both the masked residues and magnitude of added noise will be determined again, in order to allow the model to see complexes with more diversities during training. The numbers of epochs and early stopping steps were set to 100 and 30, respectively.

For the implementation of the GEE encoder, Pytorch (Paszke et al., 2019) with a default random seed 1234 was employed, and it was trained on a configuration of one NVIDIA A100 GPU.

5.3.4 The prediction of downstream PPI $\Delta\Delta G$

After pre-training, the GEE encoder has learnt how to provide an effective structural representation of a protein-protein complex with good generalization, which can be used for the downstream $\Delta\Delta G$ prediction task (figure 5.2 (B)). Specifically, for one PPI AA mutation sample point, we first constructed its KNN and radius contact graphs for the WT and MT 3D complex structures (produced by molecular generation tools), respectively. After that, the interface residues under the WT status were found based on the interfaceResidue function in PyMOL (DeLano et al., 2002) (detailed in appendix C.16), denoted as $S^{interface}$, and we marked current mutation residues as $S^{mutation}$. Next, the contact graphs of the WT and MT complexes are sent to the trained GEE encoder, to acquire two groups of residue representations for the whole WT (denoted as R^{WT}) and MT (denoted as R^{MT}) structures, respectively.

After acquiring R^{WT} and R^{MT}, following (Liu et al., 2021), we leveraged gradient-boosting trees (GBT) as the final $\Delta\Delta G$ decoder or predictor due to its good capability to effectively handle high-dimensional features and overcome overfitting under relatively small datasets. To get the informative input of GBT, we generated the following representations for comparing the geometric difference between WT and MT structures from different perspectives:

- Max-pooling and mean-pooling of representations of mutation sites in WT complex: $max_pooling(R^{WT} \in S^{mutation})$ and $mean_pooling(R^{WT} \in S^{mutation})$.

- Max-pooling and mean-pooling of representations of interface sites in WT complex: $max_pooling(R^{WT} \in S^{interface})$ and $mean_pooling(R^{WT} \in S^{interface})$.
- Max-pooling and mean-pooling of representations of mutation sites in MT complex: $max_pooling(R^{MT} \in S^{mutation})$ and $mean_pooling(R^{MT} \in S^{mutation})$.
- Max-pooling and mean-pooling of representations of interface sites in MT complex: $max_pooling(R^{MT} \in S^{interface})$ and $mean_pooling(R^{MT} \in S^{interface})$.
- The difference between the mutation sites in WT and MT complexes: $max_pooling(R^{WT} \in S^{mutation}) - max_pooling(R^{MT} \in S^{mutation})$ and $mean_pooling(R^{WT} \in S^{mutation}) - mean_pooling(R^{MT} \in S^{mutation})$.
- The global geometric representation of MT complex: $mean_pooling(R^{MT})$.

Above representations will be concatenated together and sent to the GBT decoder, and the output of GBT is the final predicted PPI $\Delta\Delta G$ representing the binding affinity or binding free energy change from wild-type to mutant status (i.e., $\Delta G^{WT} - \Delta G^{MT}$). In addition, for the selection of GBT hyper-parameters, the learning rate, number of sub-estimators, and maximum depth of sub-estimators are set to 0.001, 5×10^4, and 6, respectively. The implementation of this GBT was based on the scikit-learn library (Pedregosa et al., 2011). An overall summary of tools for implementing MpbPPI is provided in appendix C.17.

5.4 Chapter Summary

This chapter mainly describes a flexible framework for predicting the effect of amino acid mutations on protein-protein complexes (structures), which enables the screening of (similar) candidate binders sharing stronger binding affinities to the target protein. The proposed framework MpbPPI captures the binding difference between wild-type protein and mutants based on a pre-trained equivariant GNN, which fully learns the input structures' interaction information between residues, between side chain atoms, and between side chain and residue backbone atoms. The experimental results on four amino acid mutation datasets based on two different experimental

settings over three types of mutant sources display the promising performance of MpbPPI, which validates the reasonableness of our design.

Chapter 6

MCGLPPI: Integration of molecular coarse-grained model into geometric representation learning for protein-protein complex property prediction

Within this chapter, we try to explore that, in addition to commonly-used atom-scale or residue-scale protein graph representations, whether there are other ML-based methodologies for GRL techniques, which could fuse the benefits from the both scales for general protein (or complex) candidate screenings serving early drug discovery. Motivated by this, we aim to develop a mesoscopic-scale (the scale between microscopic atom-scale and macroscopic residue-scale) GNN representation learning framework functioning between the atom- and residue-scale based on the effective biological assumptions, and validate it on different protein screening tasks to test the generalization capability. Section 6.1 introduces the biological assumptions underlying our design. Section 6.2 provides the illustration of the overall framework design and the fundamental proof-of-concept experimental results that support our motivation. Besides, section 6.3 and section 7.1.4 further give the methodology details and extra extension discussion of the devised framework. Finally, section 6.4 summarizes the main content of this chapter.

The main contents (including figures) of this chapter have been published in *Nature Communications* with the DOI: https://doi.org/10.1038/s41467-024-53583-w (Yue et al., 2024).

6.1 Introduction

Protein-protein interactions (PPIs) play a pivotal role in regulating diverse cellular processes, including signal transduction, immune response, and metabolic regulations (Wang et al., 2024; Yue et al., 2023c). Gaining insights into PPI aids in understanding protein functions and identifying potential drug targets (Chen et al., 2014; Yue et al., 2023b; Liu et al., 2007; Wang, Cang, and Wei, 2020). While traditional experimental techniques for studying PPIs, such as yeast two-hybrid screening (Koegl and Uetz, 2007), co-immunoprecipitation (Lin and Lai, 2017), pull-down assays (Louche, Salcedo, and Bigot, 2017), and fluorescence resonance energy transfer (FRET) (Hussain, 2009), are effective, they often require extensive labor and substantial financial investment. To address these challenges, advancements in computational tools and artificial intelligence (AI) algorithms have transformed the study of PPIs (Peng et al., 2023). These in-silico strategies leverage expansive datasets to predict PPIs, enabling interaction site prediction (Zhou and Qin, 2007), interaction type classification (Louche, Salcedo, and Bigot, 2017), and binding affinity prediction (Wang et al., 2005).

The three-dimensional (3D) structures of proteins are fundamental to their biological functions (Alberts, 2002; Whisstock and Lesk, 2003; Zhou et al., 2024). To gain a comprehensive understanding of the biological significance and detailed mechanisms underlying PPIs, decoding the geometry of protein complexes (defined in section 2.3.1) have become essential (Wang et al., 2024). Among various computational methods, Graph neural networks (GNNs) (Wang et al., 2005; Kipf and Welling, 2016) stand out with their proficiency in handling the 3D structures of proteins. By integrating spatial information and topological data inherent to protein complexes, GNNs provide a robust framework for illuminating the multifaceted nature of protein interactions (Réau et al., 2023; Townshend et al., 2020). For instance, (Jing et al.,

2020) proposed a GNN framework, GVP-GNN, which preserves rotation equivariance of protein rigid motions when capturing geometric representations of protein-protein complexes. (Zhang et al., 2022) designed a line-graph-augmented message passing scheme to inject the relative positional information between two interactive edges for different PPI prediction tasks, such as protein-protein interface identifications.

Notably, in GNNs-based methods, proteins are represented as graph structures, with nodes corresponding to either heavy atoms (i.e., the atom-scale model) or amino acid residues (i.e., the residue-scale model) (Zhang et al., 2022; Zhang et al., 2024). However, each approach has its own trade-offs. Atom-scale models, while detailed, demand extensive computational resources to manage thousands of nodes, limiting their application to large PPI systems. On the other hand, residue-scale models are more computationally tractable but may overlook critical binding details that influence specificity and affinity. To address these limitations, multi-scale information can be integrated into the node features and edge connections. However, such integration requires intricate information exchange across scales, maintaining model consistency and physical relevance, which can complicate the design process. Additionally, in both atom- and residue-scale models, edges typically represent interactions based on sequential threshold (e.g., a sequential residue distance in primary sequence of 2) or geometric distance (e.g., 10Å), aiming to capture the complex relationships between protein structures and functions. Nevertheless, using such criteria to define connections may misrepresent chemical bonds, potentially affecting predictive accuracy.

A potential solution to these issues is to adopt coarse-grained (CG) modeling, which is a well-established framework in protein molecular dynamics (MD) simulation, designed to effectively strike a balance between maintaining essential molecular details and enhancing computational efficiency. CG-scale representation simplifies groups of atoms into single sites, such as amino acid side chains or specific chemical groups. The MARTINI model (De Jong et al., 2013; Souza et al., 2021), a widely recognized CG-scale model in protein MD simulation, represents an average of four heavy atoms and their associated hydrogens with a single CG bead. It classifies beads into multiple main physical types, including polar (P), nonpolar (N),

apolar (C), and charged (Q), etc., with subtypes based on hydrogen bonding capabilities or polarity. In addition to the various bead types, the model includes numerous chemical-plausible interaction parameters, both bonded (bonds, angles and dihedrals) and nonbonded, to directly and accurately reflect the partitioning free energy of amino acid sidechains (Souza et al., 2021; Monticelli et al., 2008). Through this strategy, the MARTINI model retains essential molecular interaction features while significantly reducing computational demands. It has been successfully applied and evaluated in many PPI-related studies, including the dimerization of the amino acid side chains (Jong, Periole, and Marrink, 2012), interactions involving membrane proteins such as glycophorin A (Sengupta and Marrink, 2010), G protein-coupled receptor (GPCR) rhodopsin (Periole et al., 2012; Lamprakis et al., 2021), and the Epidermal Growth Factor Receptor complex (Lelimousin, Limongelli, and Sansom, 2016), as well as soluble protein complex interactions (Lamprakis et al., 2021) like insulin, Ras-Raf, and Barnase-Barstar. Additionally, the MARTINI force field has been integrated into the HADDOCK framework (Roel-Touris et al., 2019), enhancing its capability to predict the 3D structures of protein interactions.

Although the CG-scale offers improved efficiency, its simulations still consume more resource than PPI predictions using AI techniques. Previous efforts to integrate the CG-scale model with machine learning (ML) or deep learning (DL) methods have primarily focused on optimizing force field potential parameters, predicting the peptide self-assembly shapes, and converting the CG-scale model back to atomistic structures (Arts et al., 2023; Wang, 2022). However, a comprehensive approach that combines AI and CG modeling to predict PPI properties remains an under-explored area.

In this study, we present MCGLPPI, a lightweight geometric representation learning framework that combines GNNs with the MARTINI CG-scale models to predict the overall properties of PPI complexes. Designed to optimize computational efficiency without compromising prediction accuracy, MCGLPPI employs a specially-designed CG-scale complex graph, which maps each CG bead of protein

complexes to nodes and utilizes chemical-plausible MARTINI force field bond parameters as edges for efficient structural characterization. Additionally, we introduce a GNN-based CG geometric-aware encoder to extract the high-quality representations from the devised graph.

Our extensive validation demonstrates that MCGLPPI achieves competitive performance on multiple curated overall property prediction benchmarks for PPI structures, including binding affinity relevant prediction and interaction type classification tasks. When compared to atom-scale and residue-scale counterparts, MCGLPPI significantly improves computational efficiency, and reduces graphics processing unit (GPU) usage and total running time by more than threefold without compromising accuracy. Moreover, proteins are intricate molecular machines that typically consist of multiple domains. Domain-domain interactions (DDIs) are critical subsets of PPIs, where the interaction typically occurs between domains rather than the entire proteins (Mosca et al., 2014; Alborzi et al., 2021; Yellaboina et al., 2011). We demonstrate that the CG-scale pre-training based on DDI patterns effectively enhances the model's ability to predict PPI binding affinities. Overall, MCGLPPI emerges as a general, accurate, and efficient method for predicting PPI properties, offering a pathway to sophisticated analysis of biomolecular interactions.

6.2 Results

6.2.1 Overview of the proposed CG-scale complex geometric learning framework

We integrate biomolecular CG structures, force field parameters, and geometric-aware GNNs within the MCGLPPI framework for efficient prediction of overall properties of protein-protein complexes, which consists of three major components: (1) CG-scale complex graph generation, (2) CG-scale geometric representation learning, and (3) DDI-based CG-scale graph encoder pre-training. A comprehensive overview of the framework and its components is provided in figure 6.1.

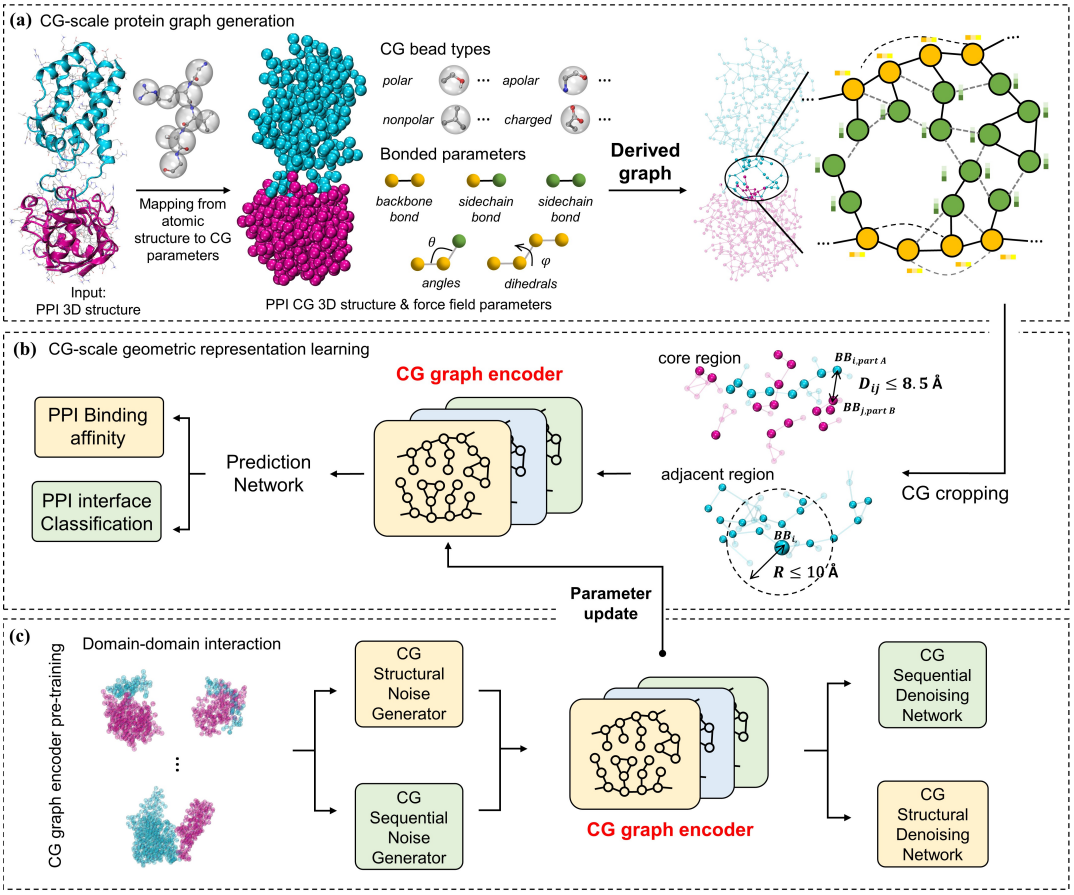


FIGURE 6.1: The flowchart of the MCGLPPI framework. **a,** CG-scale protein graph generation. The atomistic structure of a protein-protein complex is transformed into its coarse-grained (CG)-scale structure and force field parameters using the MARTINI engine (MARTINI22 and MARTINI3 are supported in this study), with these parameters encompassing bead types and bonded interactions, which further include bonds, angles, and dihedrals. Based on these structural details and parameters, we create a CG-scale protein graph where beads are represented as nodes. Bonds between these beads are represented as edges in the graph, and information on angles and dihedrals is encoded as node features **b,** CG-scale geometric representation learning. The specifically designed CG-scale protein complex graph, containing comprehensive information of MARTINI beads and bonds, is firstly cropped to identify its core interaction region. The geometric representation of this cropped graph is extracted by the corresponding CG graph encoder, which is then fed into a protein prediction network to predict the overall property of the complex. **c,** DDI-based CG-scale graph encoder pre-training. The graph encoder can be better initialized by the CG-scale pre-training techniques applied to our carefully screened (domain-domain interaction) DDI dataset. After pre-training, the graph encoder with updated model parameters can be fine-tuned to generate geometric representations with potentially more powerful capability for downstream prediction tasks.

6.2.2 Force field parameter and CG-scale complex graph generation

Structure-based prediction of PPI complex properties typically demands high-quality learning of protein geometric graph representations. The number of graph nodes and edges significantly affect computational cost. At the same time, it is crucial to ensure that the graph structure is chemically-plausible, as it is essential for accurately depicting the properties of protein complexes.

On top of this, we introduce the CG scale-based MARTINI parameterization that aims to efficiently achieve a balanced representation between chemically-plausible interaction characterization and computational cost. This process commences by transforming an atomistic PPI structure into the CG-scale structure and a comprehensive set of CG-scale force field parameters tailored for the MARTINI model (the extensively-used MARTINI22 (De Jong et al., 2013; Marrink et al., 2007) and the latest MARTINI3 (Souza et al., 2021) are examined, a crucial difference is that MARTINI3 brings richer bead types and bead numbers to slightly increase the bead resolution, and their detailed version characteristics are in appendix D.1). This simplification reduces the high-resolution atomic model into a computationally easier-to-execute form by grouping multiple atoms into fewer representative beads. The resulting parameters describe how these beads interact with each other chemically and physically from different perspectives (figure 6.2).

After integrating the structural data with the force field parameters, a multi-relational graph corresponding to the protein complex is constructed (figure 6.1 (a) and figure 6.2). Within this graph, each bead, representing a group of heavy atoms, becomes a node. The bonds between backbone beads (B) or between sidechain (S) and either sidechain or backbone beads defined by their type and length, are translated into edges that connect these nodes. It is worth noting that these nodes and edges are concise (i.e., their total numbers required to depict a protein complex are relatively lower, the corresponding statistics and further analysis for their effect on saving calculation overhead are provided in table D.1), ensuring efficient protein modeling while maintaining chemical accuracy.

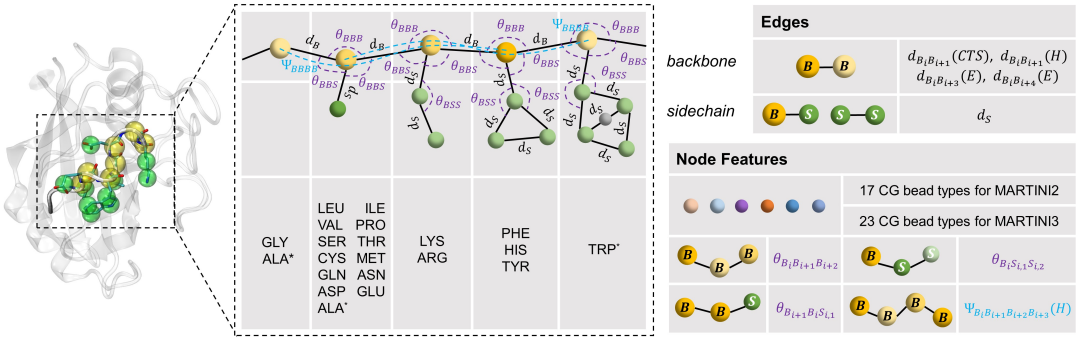


FIGURE 6.2: MARTINI-based CG-scale representation of protein structure. Each residue is represented by one backbone bead (B) and zero to five side-chain beads (S), depending on the residue type (left) and MARTINI version. The bonding interaction parameters within and between amino acids are shown (to effectively depict their local interaction structures), including bonds (types and lengths), angles (formed by triples of beads), and dihedrals (for quadruplets of beads). The right panel provides an in-depth view of the constructed CG-scale complex graph based on these efficient parameters, showcasing both the edge connections and the node features. It should be noted that in MARTINI2 CG model, ALA does not have side chain beads, whereas in MARTINI3, it has one side chain bead. TRP in MARTINI2 includes four side chain beads, while in MARTINI3, it includes five side chain beads.

Within the MARTINI framework, the protein's secondary structure plays a pivotal role in determining the bead types and associated bond, angle, and dihedral parameters for each residue. For instance, specific bond types such as constraint bonds $d_{B_iB_{i+1}}(H)$ or long harmonic bonds $d_{B_iB_{i+3}}(E)$ and $d_{B_iB_{i+4}}(E)$ are used for regions designated as helices (H) or extended strands (E), while other backbone bond parameters $d_{B_iB_{i+1}}(CTS)$ are adopted for irregular secondary structures such as coils, turns, and bends. In our CG-scale complex graph, the edge types also reflect these distinctions, facilitating the accurate description of secondary structural features within the protein complex. Furthermore, two distinct edge types, d_{intra} and d_{inter}, are introduced for the differentiation of bead nodes originating from the same or different amino acid residues, providing valuable hierarchical geometric information regarding the spatial arrangement relationships and interactions within and between the residues. Additionally, other crucial force field parameters, such as bead types, bond angles, and dihedrals, are encoded as node features within the graph (as illustrated in figure 6.2). These features are essential for capturing the spatial orientation and potential movements of the protein segments.

Furthermore, when MARTINI generates force field parameters of bond lengths, angles, and dihedrals, what it provides include the bead composition (i.e., from which beads these bonds and angles will form) and values for these bonds lengths

and angles. These values are not derived from the corresponding real conformations, instead, they are from the statistical values over samples in the Protein Data Bank (PDB) (Monticelli et al., 2008; Burley et al., 2017) database. To make these values specific to individual parameters for accurate CG-graph construction, we recalibrate them based on the actual coordinates and give them appropriate feature assignment. Please refer to section 6.3.3 for further details of defining the aforementioned multi-relational graph.

6.2.3 CG-scale geometric representation learning

To reduce the computational overhead and preserve the integrity of the data for different PPI structures, we implemented a residue backbone distance-based duel-strategy approach to graph cropping on the CG-scale complex graphs derived earlier [figure 6.1 (b)]. The initial strategy, core region cropping, focuses on the extraction of interaction interface between two proteins (or defined interaction parts of the structures beyond dimers), to ensure the focus on the most critical region of the interaction, likely enhancing model prediction accuracy and relevance. While the second strategy involves an adjacent region cropping scheme for capturing peripheral but potentially significant structural information like essential spatially correlated motifs surrounding the core interface. Through these strategies, we can produce a graph that balances detailed structural information retention with computational feasibility, regardless of the interaction pattern. The specific cropping details are in section 6.3.3.

We then apply the cropping method to each complex sample in our curated downstream datasets, which include two types of binding affinity-related regression and one interface type classification tasks. These tasks span a range of complexes, from the formation of simple dimers to the binding of the T cell receptor (TCR) to an antigenic peptide presented by the major histocompatibility complex (pMHC) (Borrman et al., 2017; Rudolph and Wilson, 2002).

We subsequently utilize a multi-relational heterogeneous GNN-based CG graph encoder (Zhang et al., 2022), which can efficiently encode the complex relationships between graph nodes and edges (detailed in section 6.3.4) within the cropped graph for generating its high-quality geometric representation. This representation is then

forwarded to the task-specific prediction network, enabling us to obtain accurate predictions of the corresponding complex overall properties.

6.2.4 DDI-based CG-scale graph encoder pre-training

Domains are fundamental structural units within proteins that are often responsible for specific functions. They play a critical role in mediating interactions with other proteins (Mosca et al., 2014; Alborzi et al., 2021; Yellaboina et al., 2011), whether within a single multifaceted protein (intra-protein interactions) or between two distinct proteins (inter-protein interactions). Despite the limited availability of detailed yet labelled 3D structural data for PPIs, the wealth of DDI structural information provides a valuable opportunity for enhancing computational models through pre-training. To this end, we use the Three-Dimensional Interacting Domains (3DID) database (Mosca et al., 2014) to construct a dataset tailored for pre-training our CG-scale graph encoder. The detailed curation process is described in section 6.3.1.

We employ a denoising-based, self-supervised pre-training approach, adapted from the work by (Zhang et al., 2024), to instruct our CG graph encoder on the intricate patterns of DDI structures and sequences. This method involves introducing perturbation to each CG graph in the pre-training DDI dataset and then forcing the encoder to reconstruct the original graph information, thereby imprinting the fundamental characteristics of domain interactions (see figure 6.1 (c) and section 6.3.5 for more details). Following this pre-training phase, the encoder, now enriched with the knowledge from the DDI dataset, undergoes fine-tuning to tackle downstream PPI prediction tasks. Through this fine-tuning process, the encoder applies the principles of domain interactions learned during pre-training to downstream PPI scenarios, potentially enhancing its ability in making predictions.

6.2.5 MCGLPPI saves computational cost while keeping competitive performance

To validate the performance and computational cost of the proposed MCGLPPI framework on the PPI complex overall property predictions, we first curated three datasets: 1) the strict protein-protein dimer subset of the PDBbind dataset (Wang

et al., 2004) (PDBbind-strict-dimer dataset), 2) the ATLAS dataset (Borrman et al., 2017), and 3) the MANY/DC dataset (Baskaran et al., 2014; Duarte et al., 2012). The former two datasets were used to evaluate the model's regression capabilities (protein-protein binding affinity predictions), while the MANY/DC dataset was used to assess the overall classification performance (protein complex interface classifications). The Pearson's correlation coefficient (R_P), root mean square error (RMSE), and mean absolute error (MAE) were utilized to assess the quality of regressions. The area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR) were for checking the capability of classification. To ensure a fair comparison, we performed the same aforementioned complex graph cropping function for each sample in every dataset (across different scales) to identify the sample's core interaction regions. Additionally, we managed to 1) compare the performance of MCGLPPI supported by MARTINI22 (denoted as MCGLPPI-M2) and MARTINI3 models (denoted as MCGLPPI-M3) and 2) further extend MCGLPPI into handling protein-protein binding affinity change regressions requiring pairwise complex structures as the input (table D.2).

6.2.6 The binding affinity prediction of formation of strict dimers

We successfully extracted protein-protein complexes exhibiting strict dimer structures from the PDBbind dataset (Wang et al., 2004). Following sample correction and label unification (i.e., converting the binding affinity labels of all relevant samples to ΔG) (Wan et al., 2022), we obtained 1270 dimer samples with binding affinity labels ΔG, referred to as the PDBbind-strict-dimer dataset (the detailed curation process can be found in appendix D.4). The standard tenfold cross-validation (CV) strategy was used to evaluate the model, specifically, aforementioned sample points will be uniformly split into 10 folds for the CV, in each iteration one fold is selected as the test set, and the rest of folds are treated as the training set. To ensure a fair comparison of the model performance across different scales, we compared the atom- and residue-scale versions of our employed protein graph encoder, GearNet-Edge (Zhang et al., 2024), using their default model settings (i.e., settings related to protein graph construction and geometric encoder hyper-parameters). Besides, we considered an atom-scale state-of-the-art geometric encoder, GVP-GNN (Jing et al.,

TABLE 6.1: Test performance and computational cost of different approaches at different scales on the 915-subset of the PDBbind-strict-dimer dataset based on one A100 GPU 40GB. The default model settings of atom-scale GearNet-Edge (here denoted as GearNet-Atom), residue-scale GearNet-Edge (denoted as GearNet-Res), and (atom-scale) GVP-GNN were adopted from (Zhang et al., 2024; Zhang et al., 2022; Jing et al., 2021), respectively.

Batch size = 128	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.584	2.103	1.623	**22,098**	**9724**
MCGLPPI-M3	CG	**0.603**	**2.083**	**1.601**	25,131	10,812
GearNet-Atom	Atom	N/A	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	N/A	OOM	N/A
Batch size = 64	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.597	2.065	1.595	**11,366**	**11,228**
MCGLPPI-M3	CG	**0.603**	**2.063**	**1.581**	13,337	12,987
GearNet-Atom	Atom	N/A	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	N/A	OOM	N/A
Batch size = 32	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.587	**2.100**	**1.619**	**5969**	**12,247**
MCGLPPI-M3	CG	**0.590**	2.109	1.621	6850	13,159
GearNet-Atom	Atom	0.527	2.320	1.788	30,597	35,993
GearNet-Res	Residue	0.578	2.182	1.664	21,295	35,336
GVP-GNN	Atom	0.436	2.299	1.795	24,639	44,174
Batch size = 16	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.585**	**2.085**	**1.615**	**3381**	**12,817**
MCGLPPI-M3	CG	0.583	2.108	1.632	3888	13,484
GearNet-Atom	Atom	0.541	2.283	1.755	17,229	47,169
GearNet-Res	Residue	0.578	2.149	1.642	10,544	38,703
GVP-GNN	Atom	0.428	2.369	1.839	13,424	46,155
Batch size = 8	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.571	**2.109**	1.636	**1853**	**15,796**
MCGLPPI-M3	CG	0.559	2.144	1.652	2230	17,355
GearNet-Atom	Atom	0.541	2.372	1.821	8476	50,545
GearNet-Res	Residue	**0.574**	2.110	**1.624**	7140	43,382
GVP-GNN	Atom	0.424	2.383	1.853	7199	48,355

1

2021), specifically designed for solving 3D macromolecular structures, particularly protein-protein complexes. Detailed information on the default hyper-parameters of all methods can be found in appendix D.5.

Furthermore, to comprehensively quantify the cost of these approaches under limited lightweight computational resources, we utilized a single NVIDIA A100 GPU 40GB to run the comparative experiments. For each approach, based on the same epoch number of 150, starting with a batch size of 8 and gradually increased it by a factor of 2 until the GPU was out of memory (OOM), and we recorded the corresponding evaluation metrics, memory usage, and total time cost across the aforementioned tenfold CV.

For the atom- and residue-scale GearNet-Edge, 915 of 1270 samples were successfully identified. To ensure a fair comparison, this 915-subset of the PDBbind-strict-dimer dataset was first used for the comparative experiment. table 6.1 presents the corresponding results. Under current experimental conditions, the key findings were as follows: (1) MCGLPPI outperformed its atom- and residue-scale counterparts. (2) Under the same batch size, MCGLPPI reduced GPU consumption by approximately 5× and 3×, as well as total elapsed time by 3× and 3×, compared to the atom- and residue-scale models, respectively, while maintaining competitive performance. These findings demonstrated the effectiveness and feasibility of introducing the MARTINI-based CG-scale representation to achieve a better balance between performance and computational cost. Additionally, MARTINI3-based MCGLPPI had moderately better performance under the best batch size (64) compared to that from MARTINI22, while with slightly increased computation overhead due to the expanding bead types and numbers.

The experimental results of MCGLPPI-M2 and MCGLPPI-M3 on the complete PDBbind-strict-dimer dataset were 0.590/0.583 (R_P), 2.071/2.067 (RMSE), 1.602/1.566 (MAE), 11,560/13,311 (GPU (MB)), and 14,312/17,446 (Time (s)). Besides, to validate the robustness of our proposed MCGLPPI, we conducted further investigations into 1) the impact of hyper-parameter, such as hidden feature dimensions on overall performance (appendix D.6), 2) more challenging training and test scenario with further structural homology reduction (table D.3), 3) model stability test using AlphaFold-generated structures (table D.4). These investigations consistently supported the conclusions outlined above.

6.2.7 The effectiveness of MCGLPPI on more complex PPI patterns

To further investigate the effectiveness of MCGLPPI in handling more complex PPI structures beyond standard dimers, the ATLAS dataset (Borrman et al., 2017), which contains the TCR-pMHC structures formed in the cell-mediated immunity processes

[1]The bold data signifies the best experimental result under the current batch size and evaluation metric. The data with the underline represents the best predictive performance according to the current evaluation metric across different batch sizes. OOM: out of memory for one A100 40GB/40,000MB GPU. N/A: not applicable.

TABLE 6.2: Test performance and computational cost of different approaches at different scales on the 451-subset of the curated ATLAS dataset based on one A100 GPU 40GB. The default model settings of atom-scale GearNet-Edge (here denoted as GearNet-Atom), residue-scale GearNet-Edge (denoted as GearNet-Res), and (atom-scale) GVP-GNN were adopted from (Zhang et al., 2024; Zhang et al., 2022; Jing et al., 2021), respectively.

Batch size = 128	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.807**	**1.084**	**0.833**	**26,358**	**5578**
MCGLPPI-M3	CG	0.803	1.121	0.861	31,514	6378
GearNet-Atom	Atom	N/A	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	N/A	OOM	N/A
Batch size = 64	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.825	1.054	**0.782**	**13,285**	**5915**
MCGLPPI-M3	CG	**0.832**	**1.013**	0.783	15,919	6500
GearNet-Atom	Atom	N/A	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	N/A	OOM	N/A
Batch size = 32	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.820	1.061	0.778	**6796**	**5992**
MCGLPPI-M3	CG	**0.825**	**1.009**	**0.756**	8066	7281
GearNet-Atom	Atom	0.813	1.109	0.851	31,018	21,375
GearNet-Res	Residue	0.818	1.068	0.796	21,059	19,866
GVP-GNN	Atom	0.524	1.599	1.169	26,232	27,231
Batch size = 16	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.821	1.042	0.788	**3524**	**7225**
MCGLPPI-M3	CG	**0.831**	**1.017**	**0.757**	4251	8034
GearNet-Atom	Atom	0.816	1.062	0.772	16,155	22,579
GearNet-Res	Residue	0.820	1.056	0.816	11,314	22,276
GVP-GNN	Atom	0.534	1.494	1.111	13,211	30,027
Batch size = 8	Scale	R_P	RMSE	MAE	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.818	1.048	0.801	**1892**	**8790**
MCGLPPI-M3	CG	0.812	**1.046**	**0.773**	2345	9447
GearNet-Atom	Atom	**0.823**	1.106	0.813	8255	26,810
GearNet-Res	Residue	0.809	1.086	0.832	6190	24,157
GVP-GNN	Atom	0.480	1.573	1.170	6946	32,253

2

along with their corresponding binding affinity values, was considered. After removing invalid samples, correcting samples, and unifying labels, we obtained 531 different structures with the ΔG labels. Please note that we utilized the structures that were optimized using the fixed backbone design option of Rosetta (Das and Baker, 2008), which were reported to achieve high structural accuracy (Borrman et al., 2017).

We performed the aforementioned standard tenfold cross-validation using the same experimental settings as the previous section and documented the corresponding evaluation results. Furthermore, the comprehensive comparison experiments were carried out based on 451 of the 531 curated samples that could be effectively processed by GearNet-Edge at both the atom- and residue-scale.

Table 6.2 shows the predictive performance and computational cost derived from the tenfold cross-validation performed on the 451-sample ATLAS subset. Additionally, we reported the best-performing results of MCGLPPI-M2 and MCGLPPI-M3 on the complete curated ATLAS dataset: 0.809/0.823 (R_P), 1.116/1.053 (RMSE), 0.837/0.803 (MAE), 13,615/16,108 (GPU (MB)), and 6982/7915 (Time (s)). Notably, when dealing with more complex protein-protein structures beyond standard dimers, the proposed MCGLPPI maintained the competitive performance and exhibited a relatively lower computational cost compared with its atom- and residue-scale counterparts, which further validated the effectiveness of the devised CG-scale protein complex geometric model and corresponding cropping function. An additional investigation into the necessity of the cropping function is conducted in section 6.2.11.

6.2.8 The prediction results for protein-protein interface classification

In addition to the aforementioned two regression tasks, an overall interface classification task for protein-protein complex was incorporated to further examine the generalizability of MCGLPPI. Specifically, the MANY (Baskaran et al., 2014) and DC (Duarte et al., 2012) datasets were utilized, containing 5739 and 161 dimers respectively. These dimers are categorized into two overall types: dimers with biological or crystal interfaces (Xu and Dunbrack, 2010). Based on this classification, the model was trained to distinguish between the two interface types, which was further formulated as a binary complex graph classification task. Following the previous data splitting convention (Wang et al., 2024; Réau et al., 2023), 80% samples of MANY, 20% samples of MANY (for MANY dataset splitting, the balance between positive and negative samples were maintained), and the complete DC datasets were used as the training, optional validation, and test sets for model evaluation, respectively.

The experiment settings from the previous two sections were kept (except for the unified epoch number changing from 150 to 30). Additionally, we compared our approach with two existing approaches, DeepRank-GNN (Réau et al., 2023)

[2]The bold data signifies the best experimental result under the current batch size and evaluation metric. The data with the underline represents the best predictive performance according to the current evaluation metric across different batch sizes. OOM: out of memory for one A100 40GB/40,000MB GPU. N/A: not applicable.

TABLE 6.3: Test performance and computational cost of different approaches at different scales on the 5535-151-subset of the MANY/DC dataset based on one A100 GPU 40GB. The default model settings of atom-scale GearNet-Edge (here denoted as GearNet-Atom), residue-scale GearNet-Edge (denoted as GearNet-Res), and (atom-scale) GVP-GNN were adopted from (Zhang et al., 2024; Zhang et al., 2022; Jing et al., 2021), respectively.

Batch size = 128	Scale	AUROC	AUPR	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.875**	**0.880**	**25,437**	**584**
MCGLPPI-M3	CG	0.873	0.855	29,172	656
GearNet-Atom	Atom	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	OOM	N/A
Batch size = 64	Scale	AUROC	AUPR	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.890**	0.871	**13,269**	**617**
MCGLPPI-M3	CG	0.882	**0.881**	15,150	704
GearNet-Atom	Atom	N/A	N/A	OOM	N/A
GearNet-Res	Residue	N/A	N/A	OOM	N/A
GVP-GNN	Atom	N/A	N/A	OOM	N/A
Batch size = 32	Scale	AUROC	AUPR	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.883**	**0.876**	**6737**	**702**
MCGLPPI-M3	CG	0.866	0.838	8248	777
GearNet-Atom	Atom	0.842	0.824	34,452	3055
GearNet-Res	Residue	0.848	0.842	24,368	3031
GVP-GNN	Atom	0.850	0.845	27,467	4132
Batch size = 16	Scale	AUROC	AUPR	GPU (MB)	Time (s)
MCGLPPI-M2	CG	0.860	0.849	**3952**	**951**
MCGLPPI-M3	CG	**0.882**	**0.864**	4417	1091
GearNet-Atom	Atom	0.850	0.799	19,290	3293
GearNet-Res	Residue	0.844	0.843	13,505	3208
GVP-GNN	Atom	0.817	0.808	15,506	4515
Batch size = 8	Scale	AUROC	AUPR	GPU (MB)	Time (s)
MCGLPPI-M2	CG	**0.881**	0.857	**2229**	**961**
MCGLPPI-M3	CG	0.877	**0.867**	2548	1065
GearNet-Atom	Atom	0.839	0.758	10,026	4017
GearNet-Res	Residue	0.870	0.859	8578	3661
GVP-GNN	Atom	0.836	0.825	8427	5194

and EGGNet (Wang et al., 2024), which had already been tested on the complete MANY/DC dataset. However, it should be noted that the effective sample numbers for atom- and residue-scale GearNet-Edge on the MANY and DC datasets were 5535 and 151, respectively. Moreover, the node feature construction in existing approaches like DeepRank-GNN relies on time-consuming external amino acid sequence alignment search, making it difficult to fairly compare computational cost. Therefore, we only compared their predictive performance on the complete MANY/DC dataset and conducted detailed computational cost comparison experiments for the atom- and residue-scale GearNet-Edge models on the 5535-151-sample subset (following the aforementioned data splitting mode).

[3]The bold data signifies the best experimental result under the current batch size and evaluation metric. The data with the underline represents the best predictive performance according to the current evaluation metric across different batch sizes. OOM: out of memory for one A100 40GB/40,000MB GPU. N/A: not applicable.

The results of the computational cost comparison experiments are shown in table 6.3. It was observed that, compared to its atom- and residue-scale counterparts, MCGLPPI achieved lower computational cost while surpassed their predictive capability. Specifically, MCGLPPI-M2 and MCGLPPI-M3 exhibited strong performance when evaluated with a batch size of 64, achieving AUROC values of 0.890 and 0.882, respectively. Additionally, the AUPR values for these models were 0.871 and 0.881, respectively. Overall, both models outperformed the performance of atom-scale and residue-scale models across different batch sizes. The reason for this improvement could be attributed to the integration of protein thermodynamics and specific secondary structure support information through the MARTINI force field, which is injected into the bonds (edges) of the CG complex graph, which provides extra distinguishable capability compared with its atom- and residue-scale counterparts. A further investigation into the importance of different CG graph edges are presented in section 6.2.10.

Furthermore, the performance of MCGLPPI supported by MARTINI22 (AUROC: 0.895, AUPR: 0.892) also outperformed both DeepRank-GNN (AUROC: 0.865, AUPR: 0.871) and EGGNet (AUROC: 0.869, AUPR: 0.863) on the complete MANY/DC dataset [the results of DeepRank-GNN and EGGNet were retrieved from (Wang et al., 2024; Réau et al., 2023)]. This observation further supported the effectiveness and generalizability of the MCGLPPI framework for predicting overall properties of PPI complexes.

6.2.9 The investigation of CG-scale pre-training techniques on different tasks

To explore the feasibility of our hypothesis that pre-training on CG-scale informative DDI complexes could benefit the downstream property predictions of CG complexes, especially in scenarios with limited labeled samples, we constructed a dataset from the 3DID database (Mosca et al., 2014), which comprises 41,663 representative DDI structures, serving as our pre-training repository. Furthermore, we implemented a CG-scale diffusion denoising-based self-supervised pre-training technique based on (Zhang et al., 2024), which enables us to capture and learn the general DDI patterns and knowledge.

Specifically, we first determined the optimal model settings for MCGLPPI under each downstream task through training-from-scratch separately. Using the same selected settings (for each task), we fine-tuned the CG graph encoder that had undergone pre-training for each respective downstream task (with the same epoch numbers as training-from-scratch). We then compared the performance difference between training-from-scratch and pre-training with fine-tuning. Moreover, to further explore the potential of the DDI dataset-based pre-training in enhancing PPI prediction models, we extended our performance comparison to the atom-scale and residue-scale (i.e., pre-training them using the original-scale corresponding pre-training settings (Zhang et al., 2024) based on the 3DID pre-training set). It should be noted that the atom- and residue-scale models (i.e., GearNet-Edge) could only recognize 33,144 out of the 41,663 DDI samples. Therefore, we selected these 33,144 samples as the common pre-training set and transformed these samples into their respective atom-, residue-, and CG-scale graphs to compare the performance across different scales (before and after pre-training) (figure 6.3 (a)).

In the PPI binding affinity prediction tasks on the PDBbind and ATLAS datasets, taking MCGLPPI-M2 as an example, pre-training improved the R_P from 0.597 to 0.606 and from 0.825 to 0.830, respectively, indicating that pre-training can effectively enhance model performance. However, for the interface type classification task on the MANY/DC dataset, the performance actually decreased after pre-training, with AUPR dropping from 0.880 to 0.866 (additional results evaluated using other metrics and those tested on the complete 3DID dataset for MCGLPPI were reported in table D.5). Meanwhile, the consistent performance change trend was found on MCGLPPI-M3, GearNet-Atom, and GearNet-Res.

We thought the contributing reasons are as follows. For the binary classification task that aims to distinguish between biological interfaces and crystal artefacts (representing non-biological interactions), training-from-scratch with current graph settings might be sufficient for the protein learning model to capture the subtle geometric structural difference between these interface types. Furthermore, pre-training based on DDI complexes extracted from actual (biological) protein-protein interactions would be more beneficial to tasks which predict the properties of complexes formed through real PPI processes (e.g., binding affinity predictions), rather than

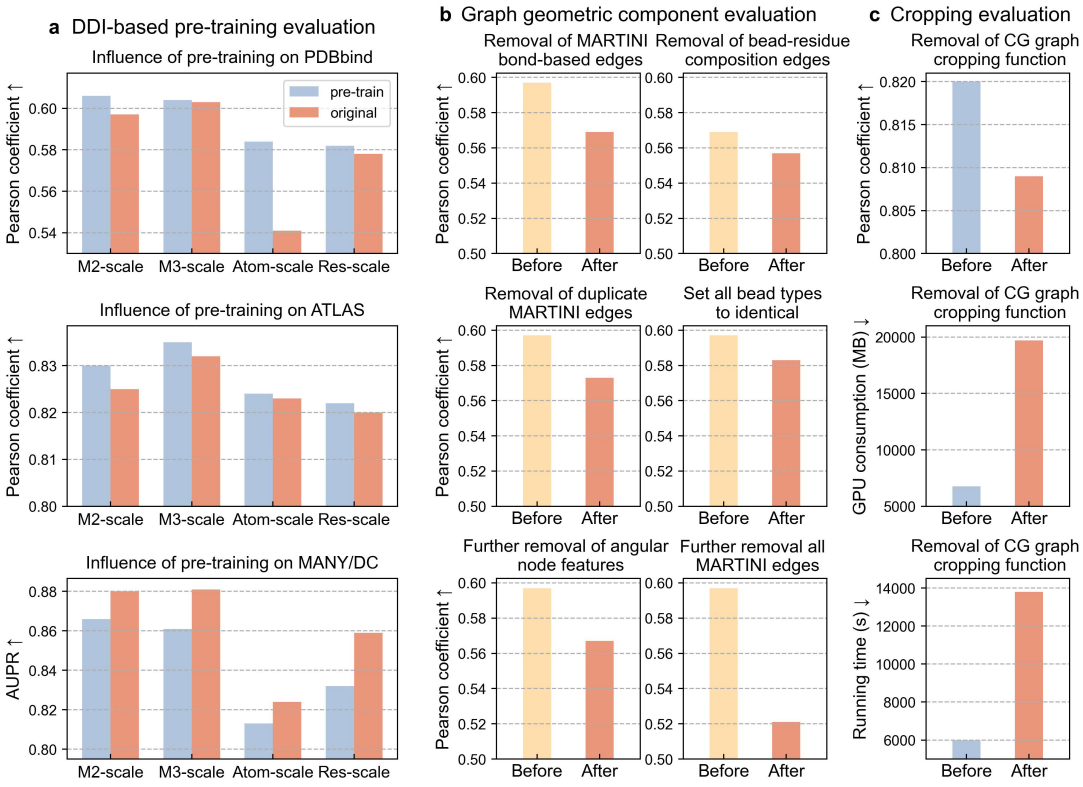


FIGURE 6.3: The investigation of performance influence contributed by the DDI-based pre-training, crucial graph geometric characterization components, and cropping function of MCGLPPI. **a,** Performance influence brought by the DDI diffusion denoising-based pre-training on the three downstream datasets. "M2-scale", "M3-scale", "Atom-scale", "Res-scale" represent the MARTINI22 CG-scale pre-training on the 3DID 33144-set, MARTINI3 CG-scale pre-training on the 33144-set, atom-scale pre-training on the 33144-set, and residue-scale pre-training on the 33144-set, respectively; and the blue and red bars display the results obtained after and before imposing the pre-training, respectively. **b,** Performance influence of removing different geometric characterization components from the devised MCGLPPI-M2 complex graph. The respective experiments about the removal of different components (based on PDBbind) are indicated on top of each sub-figure, and the yellow and red bars give the results acquired before and after removing the corresponding components, respectively. **c,** Performance influence related to the complex cropping function. Based on the maximum reachable batch size of MCGLPPI(-M2) without the complex cropping function (i.e., 32), the performance comparison results of MCGLPPI before (blue bars) and after (red bars) closing the cropping function based on the ATLAS dataset, including the R_P, GPU consumption, and total tenfold CV running time are provided. CG: coarse-grained. ↑ and ↓ represent the higher/lower the metric value, the better the evaluation performance.

distinguishing crystallographic interfaces resulting from non-biological interactions detected between repetitive crystal units (Réau et al., 2023).

In addition, we can conclude that, when using the same number of pre-training samples, our CG-scale approaches exhibited the overall better performance on all the PDBbind, ATLAS, and MANY/DC datasets. In total, pre-trainings on DDIs are effective in enhancing PPI predictions, with the CG-scale graph encoder combined with CG DDIs pre-training being the most effective approach.

6.2.10 Performance of the geometries considered in CG-scale complex graphs

To assess the impact of various geometric representations within CG complex graphs, based on the relatively simple and classical MARTINI22-based MCGLPPI, we conducted a series of ablation studies. These experiments were designed to gradually remove specific graph components and analyze their individual contributions to the overall predictive performance of the system.

Initially, we focused on edges based on chemically-plausible interactions as defined by the MARTINI(22) force field. These interactions included various types of bonds between beads, such as $d_{B_iB_{i+1}}(CTS)$, $d_{B_iB_{i+1}}(H)$, $d_{B_iB_{i+3}}(E)$, $d_{B_iB_{i+4}}(E)$, and d_S (detailed in section 6.3.3, and the following analyzed graph components can be referred from the same section). We used the subset of 915 protein dimers from the PDBbind-strict-dimer dataset for our analysis. Upon selective removal of all MARTINI bond-based edges from the CG graphs, we observed a measurable decline in performance metrics. The R_P decreased from 0.597 to 0.569 (figure 6.3 (b)), confirming the importance of these edges in accurately characterizing protein interactions.

Next, after removing MARTINI bond-based edges, we investigated the effectiveness of the proposed bead-residue geometric hierarchical composition-aware edges d_{intra} and d_{inter}. We replaced these two types of edges with the standard radius-based edges that do not differentiate the compositional relationships between chemically-plausible bead nodes and their corresponding residues (using the same edge cutoff). This modification resulted in a further decrease in predictive accuracy, with R_P dropping from 0.569 to 0.557 (figure 6.3 (b)). This emphasized the significant impact that composition-aware edges have on the model.

We also evaluated scenarios within a complete CG graph where, if both residue composition-aware edges and MARTINI bond-based edges are present between the same pair of end nodes, the MARTINI bond-based edges are disregarded. Under these conditions, the predictive performance experienced a decline from an R_P of 0.597 to 0.573 (figure 6.3 (b)). This result emphasized the importance of explicitly incorporating MARTINI bond-based edges along with the composition-aware edges in our modeling framework.

Furthermore, other groups of MARTINI force field parameters, such as bead types, angles ($\theta_{B_iB_{i+1}B_{i+2}}$, $\theta_{B_{i+1}B_iS_{i,1}}$, $\theta_{B_iS_{i,1}S_{i,2}}$), and dihedrals ($\Psi_{B_iB_{i+1}B_{i+2}B_{i+3}}$), were encoded as node or edge features within the graph. To assess their specific importance, we invalidated the bead type feature and angular features, which include (bond) angles and dihedrals, from the CG-scale graph. As shown in figure 6.3 (b), when the bead type in every node feature and edge feature were set to identical or when angular information was further omitted (on top of the former), there was a reduction in the R_P from 0.597 to 0.583 and 0.567, respectively.

Moreover, the complete invalidation of above two groups of force field parameters plus all MARTINI-based edges (i.e., only the standard radius-based edges are included), led to a further decline in R_P to 0.521 (reaching a 12.7% difference). These findings indicated that, not only does our framework rely on aforementioned chemically-plausible edges, but it also requires both the chemical and physical information provided by the MARTINI bead types and angular information from angles and dihedrals. This combined information enables MCGLPPI to make more informed predictions about the properties of protein interactions.

6.2.11 Influence of graph cropping on overall model efficiency

To provide a comprehensive analysis of the influence of graph cropping on the efficiency of the MCGLPPI framework, we conducted additional experiments using the ATLAS dataset, which contains more complex TCR-pMHC structures as a benchmark (based on MARTINI22-version MCGLPPI). We ran our model on the subset of 451 protein complex structures from the ATLAS dataset while maintaining all experimental settings consistent with our previous MCGLPPI experiments, except for the

graph cropping function, which was disabled. As shown in figure 6.3 (c), the maximum batch size that could be processed by MCGLPPI under a single NVIDIA A100 GPU 40G decreased significantly from 128 to 32 when the graph cropping function was turned off. With a batch size of 32, the R_P, GPU memory consumption, and total runtime post-cropping were 0.825, 6796 MB, and 5992 seconds, respectively, compared to 0.809, 19,706 MB, and 13,794 seconds before cropping. These results confirmed the critical role of graph cropping in improving computational efficiency and predictive performance in the framework.

6.3 Methods

6.3.1 The detailed curation process for the 3DID pre-training dataset

The latest 3DID database (Mosca et al., 2014) provides 15,983 DDI structure templates, with each template containing one or more samples of resolved 3D structural data. We removed any DDI templates from the 3DID dataset that were identical to those present in our downstream benchmark datasets. Following this stringent exclusion process, we obtained a pre-training dataset which provides 41,663 DDI structure samples in total.

6.3.2 MARTINI-based geometric parameter generation

Each complex structure was processed using pdbfixer tool (https://github.com/openmm/pdbfixer) to complete missing side-chain information and convert non-natural amino acids to their natural counterparts (for the atom- and residue-scale models, the same process was also performed). The Python-based MARTINI script martinize.py (https://cgmartini.nl/docs/downloads/tools/proteins-and-bilayers.html, version 2.4) (De Jong et al., 2013) was used to generate MARTINI22-based CG structure and force field parameters for each protein complex. The Martinize2 and Vermouth programs (Kroon et al., 2022) were for producing MARTINI3-based CG structure and force field parameters of corresponding complexes. The CG structure encompasses the bead coordinate-related information, while the force field parameters include both nonbonded parameters, such as bead types, and bonded parameters,

including bonds, angles, dihedrals, and bead connectivity instructing the bead composition for these bonds and angles (figure 6.2).

MARTINI-based bonds information

Within the MARTINI (applicable for both MARTINI22 and MARTINI3) force field representation, there are two principal types of bonds: backbone bonds and sidechain bonds. As illustrated in figure 6.1 (a) and figure 6.2, a backbone bond d_B is formed between two neighboring backbone beads (B_i). Sidechain bonds d_S occur either between a backbone bead and a sidechain bead (S_i) within the same amino acid or between sidechain beads. Furthermore, the MARTINI force field differentiates backbone bond types based on the protein's secondary structure. Consequently, d_B can be subdivided into:

1. Constraint bonds: These are formed between two adjacent amino acids that are part of a helical structure (H), denoted as $d_{B_iB_{i+1}}(H)$.

2. Long harmonic backbone bonds: For three consecutive amino acids forming extended elements (E), these bonds connect the backbone beads of residues i and $i+3$, denoted as $d_{B_iB_{i+3}}(E)$.

3. Long harmonic backbone bonds: For four contiguous amino acids in extended elements (E), the bonds connect backbone beads of residues i and $i+4$, denoted as $d_{B_iB_{i+4}}(E)$.

4. Other harmonic backbone bonds: Parameters between two adjacent amino acids for irregular secondary structures such as coils, turns, and bends are denoted as $d_{B_iB_{i+1}}(CTS)$.

In total, there are five types of bonds in both MARTINI22 and MARTINI3 force fields: $d_{B_iB_{i+1}}(CTS)$, $d_{B_iB_{i+1}}(H)$, $d_{B_iB_{i+3}}(E)$, $d_{B_iB_{i+4}}(E)$, and d_S. These bond types were chosen as the primary edge types in our MCGLPPI framework.

MARTINI-based angles and dihedrals information

The bonded parameters also include angle and dihedral parameters (figure 6.2). Specifically, there are three types of angle parameters and one dihedral type being considered:

1. $\theta_{B_iB_{i+1}B_{i+2}}$: The angle between three consecutive backbone beads.
2. $\theta_{B_{i+1}B_iS_{i,1}}$: The angle formed between a backbone bead, its neighboring backbone bead, and the first sidechain bead of the amino acid.
3. $\theta_{B_iS_{i,1}S_{i,2}}$: The angle between a backbone bead and two consecutive sidechain beads of the same amino acid.
4. $\Psi_{B_iB_{i+1}B_{i+2}B_{i+3}}$: The dihedral angle between four consecutive backbone beads. It is noted that dihedral angles were imposed only when all four interacting beads had the helical secondary structure (H) in MARTINI force field.

Notably, the bond lengths, angle and dihedral values were re-calibrated based on the given coordinates of corresponding endpoint beads. They were not directly adopted from the statistics values from PDB database since the calibration can provide accurate geometric interactive information for the CG-scale protein complex graph model.

6.3.3 The construction of CG-scale protein complex graph and its cropping function

The challenge to build an effective CG protein complex graph is how to fully preserve the introduced MARTINI parameters in a graph structure accurately and efficiently, while keeping the flexibility of injecting other useful knowledge on top of these MARTINI parameters. To overcome this challenge, we first modelled a given protein-protein complex as a multi-relational contact graph $\mathbb{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. $\mathcal{V}$ represents the set of graph nodes i, i.e., all MARTINI beads produced for the complex, and the position of each bead node i is determined by its equipped 3D coordinate. $\mathcal{E}$ and $\mathcal{R}$ are the set of edges between bead nodes and the set of edge types r, respectively. Based on this, we denoted an edge from nodes j to i with type r as (i, j, r)).

In order to give the precise descriptions of inter-bead geometric positional relationships while integrating concise chemical-plausible MARTINI bonds, the graph structure is built as follows. First, an edge will be wired if any two bead nodes have the Euclidean distance smaller than 5Å. Compared with the commonly-used radius edge used in atom-and residue-scale models, MCGLPPI further distinguishes the edge type based on whether the two end bead nodes are from the same residue. In other words, the edge is categorized as an intra-residue contact edge d_{intra} if the two bead nodes belong to the same residue otherwise is an inter-residue contact edge d_{inter}, for which the hierarchical composition information between chemical-plausible beads and their surrounding residues within a complex is injected. Besides, all bond types described in the MARTINI-based bonds information section are incorporated in the graph edge structure.

To summarize, there are seven types of edges in total, describing the protein complex geometry from different perspectives, including various precise geometric contact relationships, actual secondary structure supports, and chemical bonded interactions, etc. Although multiple types of edges are included, due to the conciseness of MARTINI bonds, the average degree of nodes in the constructed graph is still relatively small (see table D.1 for further comparison analysis with the atom- and residue-scale counterparts), which contributes to relatively low representation learning overhead. Next, the other MARTINI parameters were allocated as follows into these defined nodes and edges as their features (the definition of generation of a one-hot representation is in appendix D.12).

Bead node features $\mathbf{f}_i$

- The MARTINI22 or MARTINI3 bead type, given as a one-hot representation
- Sine-cosine encoded backbone angles ($[sin(\theta_{BBB}), cos(\theta_{BBB})]$)
- Sine-cosine encoded backbone-side chain angles ($[sin(\theta_{BBS}), cos(\theta_{BBS})]$)
- Sine-cosine encoded side chain angles ($[sin(\theta_{BSS}), cos(\theta_{BSS})]$)
- Sine-cosine encoded backbone dihedrals ($[sin(\Psi_{BBBB}), cos(\Psi_{BBBB})]$)

Graph edge features $\mathbf{f}_{(i,j,r)}$

- The one-hot MARTINI22 or MARTINI3 bead type of the source node
- The one-hot MARTINI22 or MARTINI3 bead type of the target node
- The edge type, given as a one-hot representation
- The absolute positional difference between source and target nodes in the MARTINI22 or MARTINI3 bead sequence, given as a one-hot representation
- The calibrated bond length

For the above individual node and edge features, they were concatenated as the final features [$\mathbf{f}_i : \mathbf{R} \in 1 \times 25$ (MARTINI22) or 1×31 (MARTINI3), $\mathbf{f}_{(i,j,r)} : \mathbf{R} \in 1 \times 53$ (MARTINI22) or 1×65 (MARTINI3)]. Besides, since the angular parameters generated from MARTINI are sparse (see figure 6.2, not every bead will involve in the calculation of every type of angles), an extra rule (appendix D.13) was provided, to assign sparse angular node features to specific beads to avoid potential conflicts.

After the definition of the CG-scale complex graph, the corresponding graph cropping function was designed to identify its core interaction regions for further reducing the computational cost and potentially increasing the predictive accuracy. Specifically, we first determined the interaction parts of the protein-protein complexes with different interaction patterns. For the curated complexes in the PDBbind and MANY/DC datasets, they belong to the standard dimers, we chose each protein chain as one part, thus a complex can be represented as two interaction parts. For the structures in ATLAS which usually contain 4 or 5 chains, the peptide and MHC chains were treated as the first part, while the second part contained the remained TCR chain structures.

Next, a distance matrix $\mathbf{M}^{dis}$ based on the specified two interaction parts was created to guide the generation of the cropped complex graph $\mathbb{G}^{'}$. For each residue, MARTINI will only assign one backbone bead (B) with a 3D coordinate to represent its backbone atoms and their overall position, and thus the coordinate of B was used as the position of the residue [analogous to using alpha carbon ($C\alpha$) as the residue position in residue-level protein graph constructions (Zhang et al., 2022)]. Based on

this, $\mathbf{M}^{dis}$ with the size of $L_{AA1} \times L_{AA2}$ can be calculated, in which L_{AA1} and L_{AA2} are the amino acid (AA) sequence length (given by above backbone beads B) of the interaction parts 1 and 2, respectively. In $\mathbf{M}^{dis}$, every element $\mathbf{M}^{dis}_{i,j}$ was the pairwise Euclidean distance between corresponding residues from separate interaction parts (given by B_i and B_j). Then any pair of residues having the distance smaller than 8.5Å were retained as the core region (i.e., the initial strategy). Other AAs that had $B - B$ distance to any core region AAs smaller than 10Å were also retained (i.e., the second strategy).

After that, all bead nodes and edges within the retained region were kept as the cropped complex graph $\mathbb{G}^{'}$ (the angular node features were further re-calibrated if any end bead nodes within the angular information calculations were removed by this cropping). Furthermore, for the atom-scale and residue-scale models, to conduct a fair comparison, the same cropping function were employed to their protein graphs, the only difference was replacing B with $C\alpha$ to indicate the overall position of the residue.

6.3.4 The CG-scale representation learning for complex overall property prediction

After acquiring the cropped protein complex graph $\mathbb{G}^{'}$, a representative multi-relational heterogeneous GNN-based protein encoder GearNet-Edge (Zhang et al., 2022), was incorporated into the framework, for predicting the overall property of protein complexes. Specifically, based on a line graph-enhanced edge message passing mechanism (Harary and Norman, 1960) to model the inter-edge positional relationships, the additional structural information can be injected into the node representations for more effective protein geometric interaction modeling (the corresponding equations are provided in appendix D.14). We made it work at the CG-scale for generating the overall geometric representation of the input CG cropped graph $\mathbb{G}^{'}$, and the generated representation was further learnt by a three-layer task-specific multi-layer perception (MLP) to give the final property prediction results.

6.3.5 The DDI-based CG graph encoder pre-training technique

The protein domain-domain complex parameterized by MARTINI still preserves the fundamental conformation and chain sequence, but the basic particles are substituted from original atoms to CG beads. Intuitively, performing the self-supervised noise-adding-denoising pre-training techniques, which are already demonstrated to be effective on understanding geometric regulations of proteins at atom- or residue-scale (Liu et al., 2021), could also benefit the understanding of general knowledge from CG DDI complexes (for downstream property predictions).

Therefore, based on the atom-scale work (Zhang et al., 2024), a CG-scale complex pre-training technique was developed, which adds noise with changing magnitudes into 3D coordinates and sequences of MARTINI-based CG bead nodes based on the diffusion mechanisms (Ho, Jain, and Abbeel, 2020), for the CG-complex geometric regulation learning. The equations and complete details are provided in appendix D.15. After the pre-training, the trained CG graph encoder will be fine-tuned on the specified downstream task to produce the effective structural representations for corresponding input CG complex graphs.

For the implementation of the CG-scale protein complex graph construction, representation learning, and pre-training processes, Pytorch (Paszke et al., 2019) and Torchdrug (Zhu et al., 2022) with a default random seed 0 were employed, and the Adam (Kingma, 2014) with the initial learning rate of 0.0001 was adopted as the optimizer for model training (the same environment settings were also used for the other involved models working at the atom- and residue-scale). Besides, all experiments were deployed on a configuration of one NVIDIA A100 GPU 40GB. A complete summary of tools for implementing MCGLPPI is given in appendix D.16.

6.4 Chapter Summary

This chapter introduces a verstile proof-of-concept framework, MCGLPPI, aiming to demonstrate the feasiblity of modelling GRL protein complex structures at the mesoscopic molecular coarse-grained (CG)-scale based on the parameters from CG force fields for various complex-based candidate screening tasks. The widely-used CG

force fields MARTINI22 and MARTINI3 are considered and tested. The experimental results on four main datasets covering binding affinity predictions, binding affinity change (brought by amino acid mutations) predictions, and interaction interface type predictions under different experimental evaluation settings and protein structure sources show that, compared with its atom- and residue-scale counterparts, MCGLPPI achieves a better trade-off between computational cost and chemical-plausible interaction characterization. This validates the effectiveness and feasibility of our hypothesis and framework design.

Chapter 7

Discussion

Based on reasonable biological understandings, this thesis provides a number of specialized GNN-related GRL frameworks for solving a series of candidate screening problems under different data scales in early drug discovery. After a thorough literature review of the subject in chapter 2, based on reasonable biological and chemical knowledge-driven hypotheses, we customize the drug-disease-target relationship networks, the drug-target interaction-based biological networks, dimeric protein structures, as well as more complex protein complex structures as the proxy graphs, to investigate their inherent topological and geometric characteristics, ultimately serving as the distinguishing features aimed at different downstream tasks. This present chapter now summarizes the contributions of this thesis and highlights potential future directions for all of the work proposed here following the illustration order given in chapter 2.

7.1 Contributions & Conclusions

In this section, those stated in chapter 1 and other chapters are recapped, to summarize how these early drug discovery tasks in this thesis are formulated and solved, and what insights can be given based on the experimental results.

7.1.1 Ranking of Novel Drug Targets with the Consideration of Disease Types Based on Hyperbolic Lorentz Space Representation Learning

Within chapter 3, we first hypothesized that, because heterogeneous DDT networks could possess hierarchical structures, the (translation-based) KGC method could

benefit from properly introducing hyperbolic space which is natural for representing network hierarchies. Within the scope of the more practical target identification problem – directly ranking all candidate targets for the given drug while explicitly considering disease types, we formulated this problem as a hyperbolic translation-based triplet target entity completion task, for testing our hypothesis. We proposed FLONE (fully Lorentz network embedding) and evaluated it on two curated hierarchical DDT networks. Our experimental results showed that, FLONE generates more accurate target predictions than its Euclidean counterparts, which supports our hypothesis. Furthermore, we found that external domain knowledge, such as drug structural and target sequence similarities can be utilized to further improve the predictive accuracy for both seen and previously unseen triples in our framework.

7.1.2 Screening of Drug Pairs with Better Therapeutic Effects Based on Meta-path-based Heterogeneous Network Representation Learning and Auxiliary Adverse Effect Learning

Based on the hypothesis that the therapeutic effect prediction task could benefit from adverse effect information, in chapter 4, we first formulated the therapeutic synergy score prediction as a multi-task heterogeneous network learning problem that simultaneously learns synergy score and adverse effect predictions for drug-drug pairs. To solve this problem, we proposed Muthene, a multi-task heterogeneous network learning method specifically for synergy score predictions.

To capture MoA information for the both predictions, which is critical for rational drug discovery, Muthene extracted meta-paths to represent the underlying chemical and/or molecular network interactions. Our experimental results showed that Muthene generated more accurate synergy score predictions than standard single-task learning methods, which supports our hypothesis. Our method is a rational way to predict TEs based on multiple sources of information, including utilizing AE information, which can effectively improve the TE prediction accuracy. More significantly, using a drug pair *Vincristine—Dasatinib* as a case study, Muthene generated a few interesting hypotheses about the underlying MoAs of drug combinations. For example, Muthene revealed the synergistic effects of apoptosis and necroptosis

caused by *Vincristine* and *Dasatinib*, respectively, and the additive effect exerted by LRRK2, an unexpected cancer target indirectly modulated by *Vincristine*.

7.1.3 Identification of Candidate Protein Complex Mutants with Stronger Binding Affinities Based on Pretraining-based Equivairant Geometric Representation Learning

Mutations can alter the formation of protein complexes by enhancing or inhibiting interactions between proteins. This, in turn, affects various functions in the body, including catalyzing chemical reactions, transporting molecules, and transmitting signals. Gaining insights into how mutations influence protein-protein interactions can help us understand the development and progression of diseases, ultimately leading to the creation of more effective treatments (the further example can be found in appendix C.10).

Computational modeling of changes in binding free energy ($\Delta\Delta G$) resulting from mutations allows for the prediction and manipulation of PPIs on a large scale. In chapter 5, to better learn complete geometric regulations for limited numbers of downstream PPI $\Delta\Delta G$ samples from the proper pre-training dataset, we established a novel pre-training-based computational framework MpbPPI. MpbPPI can effectively evaluate the impact of single and multiple missense AA mutations on PPI binding affinity based on more comprehensive and computationally inexpensive geometric characteristics and flexible adapt to multiple mutant generation tools, which could provide valuable insights into the potential consequences of mutations on protein-protein interactions.

Recent advances in the PPI $\Delta\Delta G$ prediction methods, particularly the geometric GNN-based methods (Liu et al., 2021; Jiang et al., 2023), have significantly improved the predictive accuracy and stability. However, many methods focus on modelling single residue-wise contact relationships or inter-atom interactive relationships, omitting more comprehensive geometric relationships between different constituents in protein-protein complexes. Our MpbPPI addresses this issue by propagating specially designed different-scale geometric interactive information between residues, between side chain atoms, and between side chain and residue backbone atoms based on a new geometric-equivariant message passing network.

In addition, the scarcity of labelled PPI $\Delta\Delta G$ data presents a significant challenge. To overcome it, based on a strictly screened pre-training set for avoiding potential information leakage, we proposed four pre-training tasks to be performed simultaneously, which are compatible with the above different-scale geometric information, to improve MpbPPI's generalization to unseen downstream samples. The robust predictive ability of the proposed geometric characteristics learning framework and corresponding pre-training tasks has been extensively validated on four benchmark datasets and compared with seven different methods under two realistic evaluation settings (the further discussion of some other optional features is provided in appendix C.12).

Furthermore, accurate prediction of PPI $\Delta\Delta G$ usually requires the availability of the PPI mutant structure, which can be generated using various methods. To satisfy this, MpbPPI was designed to flexibly fit the mutant structures from different sources. In this study, we used three tools to build mutant structures: FoldX, MODELLER, and AlphaFold2. FoldX substitutes and optimizes original residue structure based on the given wild-type complex, MODELLER generates the structure through homology modelling, and AlphaFold2 can predict the protein structure from scratch. Our research found that when MpbPPI is combined with the three mutation structure generation tools FoldX, MODELLER, and AlphaFold2, each combination achieves good prediction results. There is a strong positive correlation between the predicted and measured PPI $\Delta\Delta G$ values. Specifically, based on the ten-fold cross-validation, for the smallest single mutation dataset, each combination of tools achieved a R_P greater than 0.57. For the multiple mutation dataset, each combination of tools achieved a R_P more than 0.76. These findings further confirmed MpbPPI is a dependable and efficient method for predicting PPI $\Delta\Delta G$. Moreover, it can serve as a foundational framework and be flexibly combined with various mutation generation tools to improve the accuracy of PPI $\Delta\Delta G$ predictions.

Our results also showed that the FoldX-based MpbPPI prediction outperformed the MODELLER-based prediction, with the AlphaFold2-based prediction yielding the worst results. We observed that, compared with the WT structures, the mutant structures generated by FoldX mainly exhibited conformational changes in the

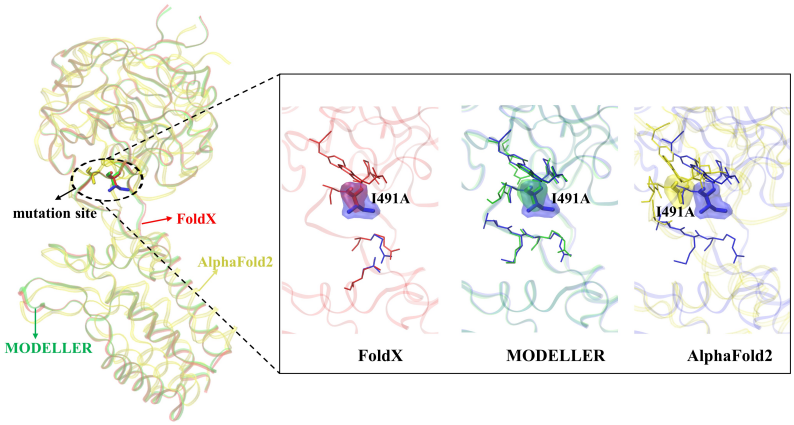


FIGURE 7.1: Comparison of mutant PPI structures from various mutant generation tools. An example of the structural differences between WT and mutant structure (PDB ID: 1AK4). The WT structure is depicted in blue, while the mutant structures generated by FoldX, MODELLER, and AlphaFold2 are shown in red, green, and yellow, respectively. The mutant amino acid and its neighboring amino acids' backbone and side chain are represented as sticks, with the mutant amino acid highlighted in surface style. The $C\alpha$ RMSD values between the WT and mutant structures were 0 Å for FoldX, 0.3 Å for MODELLER, and 3.8 Å for AlphaFold2.

side chain of the mutant residue and their neighboring residues (figure 7.1). Meanwhile, MODELLER-based mutant structures showed slight alterations in both the side chain and backbone of the mutant residue and their neighboring residues (figure 7.1). AlphaFold2 displayed the largest variations, with the $C\alpha$ root mean square deviation (RMSD) between the WT and AlphaFold2-based mutant structures ranging from 0.7Å to 40.6Å in the M1101 dataset. Among these, the mutation structures generated by AlphaFold2 exhibit larger conformational changes, providing a wider learning space for the model [AlphaFold2 does not directly generate mutant structures based on WT templates (Jumper et al., 2021)]. However, this also increases the prediction complexity, necessitating more data during training to improve the model's prediction accuracy. Besides, for FoldX and MODELLER, a mutant structure usually can be generated within $1 \sim 2$ minutes, while AlphaFold2 needs an average of dozens of minutes. Therefore, when considering the conformational changes after protein mutations, striking a balance between the prediction complexity/availability and the amount of training data is needed to achieve better prediction accuracy.

7.1.4 Screening of Protein (Complex) Candidates towards Broader Property Criteria Based on Molecular Coarse-Grained Force Field Modelling

In chapter 6, we presented MCGLPPI, an efficient framework that enhances the structure-based overall property predictions for protein-protein complexes by utilizing the MARTINI force field for lightweight protein modeling. At the mesoscopic CG-scale, our proposed CG protein graph model uses concise yet chemically plausible beads and bonds to accurately represent the conformation characteristics of protein-protein complexes. This approach results in lower computational overhead, leading to a better balance between predictive performance and cost compared to its atom- and residue-scale counterparts. Further, the modern design of protein graphs and corresponding GNN protein encoders like GearNet-Edge is more relying on the construction of edges, which are usually fully built based on multiple pre-defined geometric distance and sequential thresholds, aiming to capture more comprehensive spatial relationships between particle nodes (Zhang et al., 2022; Zhang et al., 2024; Zeng et al., 2023). While the number of edges will significantly influence the neighboring message aggregation (Kipf and Welling, 2016) speed of corresponding GNNs. Our devised CG-graph incorporates more chemical-plausible MARTINI-based edges wiring designed bead nodes pairs based on specific interaction definitions, thereby reducing the reliance on indiscriminately connecting every node pair within multiple pre-defined thresholds, which is ultimately beneficial to decreasing the processing overhead under current framework (more detailed explanation can be found in table D.1).

On top of this, MCGLPPI was designed to adapt to the geometric parameters from both the classical MARTINI22 and the more recent MARTINI3 force field for flexible extensions and comparisons. Under current experimental settings, different versions achieve better performance on different datasets, while the DDIs-based pre-training narrows down the difference between the two versions. Besides, MARTINI22-based MCGLPPI enjoys slightly quicker processing speed due to less bead types and bead numbers (please see more specific performance difference analysis across versions in appendix D.10).

In addition, our extensive ablation studies highlighted the significance of both edge and node features derived from the MARTINI force field for accurate PPI predictions using the MCGLPPI framework. Notably, these features, like chemical bond-based edges and physical-plausible bead type node features, are crucial in capturing the essential properties that govern protein interactions (The potential extension analysis of using the force field parameters in other scales can be found in appendix D.11). Through our proposed CG-scale learning framework, we also demonstrated the effectiveness of DDI-based pre-training in improving binding affinity predictions of PPIs.

7.2 Future Directions

As our literature review and experiments have revealed, there are a number of related works that can be considered for extending the work proposed in this thesis in the future.

7.2.1 Incorporate Non-Euclidean Space Embedding with Positive Curvatures for Learning Network Sub-structures

Apart from the heterogeneous DDT networks, FLONE proposed in chapter 3 could be applied to other complex heterogeneous networks with a hierarchical structure. However, it is worth mentioning that, apart from hierarchical structures, real-world complex networks could exhibit other types of sub-structures, e.g., the cyclic structure. Therefore, in future work, it is promising to introduce more non-Euclidean space, e.g., spherical space with positive curvatures specifically for learning cyclic structures into our framework, making it capable of adapting to different network sub-structures. Additionally, FLONE is a network (graph)-based method mainly utilizing known DDT network structures to infer unknown associations, therefore exploring how to give a confident prediction to samples which are totally irrelevant to known DDT triplet relationships is also an interesting future direction.

7.2.2 Incorporate More Biological Plausible Meta-path Varieties for MoA Modelling

When considering using meta-paths to model the DTI commonalities for TE learning (chapter 4), we must admit that the meta-paths are an over-simplification of the complex molecular and chemical networks, which might not fully reflect the complexity of interactions between drugs and targets. More semantically meaningful meta-paths from external data sources could be introduced to further improve the performance of Muthene in terms of the TE prediction and MoA illustration.

7.2.3 Incorporate Biochemical and Evolutionary Related Information into Geometric Representation Learning for Mutant Screening

A future direction of improving MpbPPI framework depicted in chapter 5 is to include more flexible biochemical and evolutionary information compatible with existing different-scale geometric characteristics and pre-training tasks. It is also interesting to incorporate the features like the multiple types of PPI interface information, to make the framework able to better adapt to some more specific application scenarios, for example, predicting the PPI $\Delta\Delta G$ in the case that most of the AA mutations occur at the interface of protein-protein complexes.

7.2.4 Support More Molecular Coarse-Grained Force Fields and Protein-Protein Complex Generation Tools for Broader Protein Complex Screening

While MCGLPPI (introduced in chapter 6) has shown promising overall performance, there are some areas for further improvement or investigation. For instance, MCGLPPI may not fully capture the complexity of protein-protein systems, and other CG-scale protein modeling systems deserve further exploration. Besides, the MCGLPPI model, built on a geometric GNN framework, learns from the confident 3D structures of complexes to predict related overall properties. Although with simple modifications, we further demonstrated its versatility by straightforwardly extending it to binding affinity change (i.e., $\Delta\Delta G$) calculation of pairwise wildtype-mutant complexes on an independent multiple-point amino acid mutation dataset

(detailed in table D.2), it currently lacks the capability to (directly) utilize the more broadly available and readily accessible PPI sequence data as initial input of predicting PPI attributes.

Based on this, a future plan would be to incorporate more cost-efficient geometric information to more comprehensively characterize CG complex structures, e.g., considering the Euler angles for describing the relative rotation between CG particles, and to support more CG modeling systems for capturing protein-protein thermodynamic quantities and underlying chemical mechanisms from different perspectives. There is another potential to further improve model performance on more PPI tasks by integrating sequence co-evolutionary information as a feature component (Singh et al., 2022). Additionally, combining our CG-scale framework (with further adaptive modifications) more intensively with tools that can predict confident PPI structures based on sequences, such as AlphaFold3 (Abramson et al., 2024), AlphaFold-Multimer (Evans et al., 2021), FoldDock (Bryant, Pozzati, and Elofsson, 2022), or the MARTINI force field-integrated HADDOCK (Roel-Touris et al., 2019), will open avenues for predicting PPI properties, such as determining whether two proteins interact and better understanding the effects of mutations on these interactions.

7.2.5 Incorporate Inductive Target Protein Initial Features to Increase the Generalization Capability of Models

The four main frameworks proposed in this thesis, including FLONE (chapter 3), Muthene (chapter 4), MpbPPI (chapter 5), and MCGLPPI (chapter 6), are a series of proof-of-concept GRL frameworks that are developed for demonstrating the respective underlying biological knowledge-driven hypotheses based on the modelling of topological or geometric characteristics of objective source data.

Practically and generally, to acquire a more robust capability under various actual screening application scenarios, these graph-based frameworks could further benefit from integrating inductive protein sequential features extracted from the pretrained large protein language models (PLMs) (Lin et al., 2023; Elnaggar et al., 2021) as the initial entity features.

Specifically, the PLMs which are typically trained with over millions of protein sequences in available protein databases could bring synergistic generalization capability into different downstream protein tasks, through producing token representations [e.g., amino acid (AA) embeddings] with rich sequential similarity regulation information for the input protein (Su et al., 2023). These embeddings can be used to replace or added into the original node features of constructed graphs at different scales to provide higher scalability and potentially better predictive performance.

Taking the Muthene framework as an example, when inferring a drug-drug-cell line combination where a drug is totally previously unseen from the existing DTI-PPI biological network (on which the model is trained), if the model is trained with the meta-path ego-subgraphs where node features initialized with PLM embeddings (for protein targets) and ECFP6 (for drugs), as long as when we can find/collect the meta-path ego-subgraph (i.e., the relevant drug and target entity combinations) for the new drug, the complete capability of the trained model still could be leveraged (rather than just relying on ECFP to generate the feature for this new drug – the information from the DTI biological network is missing). This is because if use the one-hot initialized drug and target node features to characterize the meta-path ego-subgraph, the total drug and target varieties would be determined in advance by the nature of the fixed-length one-hot features here, while using more inductive node features like target ones from PLMs (just requiring knowing the AA sequence in advance) and drugs ones from ECFP [only needing the string information of chemicals (e.g., SMILES) (Huang et al., 2020)] could alleviate this theoretically.

Another illustrative example is that, it could be also promising to incorporate AA embeddings from PLMs into the MCGLPPI framework (as extra concatenated initial bead node features), to make it directly propagate the CG force field information and (more high-level) AA similarity information simultaneously. This is theoretically feasible as the scale of the mesoscopic CG graphs is smaller than the AA residue scale. For the CG bead nodes belonging to the same AA, the same AA embeddings could be used for their node features after the dimension reduction.

Appendix A

Supplementary for Chapter 3

A.1 The triplet screening based on Jaccard coefficient

To avoid the implicit data leakage caused by very similar predicates, we perform the following Jaccard-similarity-based (Murphy, 1996) screening procedure: we first retrieve the drug-target pair set corresponding to every disease, and, for each disease $D_k^{'}$, we calculate its Jaccard coefficient with all other diseases based on their corresponding drug-target pair sets, if any of the Jaccard coefficients for $D_k^{'}$ are larger than 60%, this disease and corresponding known triplets are removed.

A.2 The GO-PPI network and its encoder

To construct the GO-PPI network which is used to rank disease-target (gene) pairs, similar to (Chen, Althagafi, and Hoehndorf, 2021), we first obtain gene ontology (GO) annotations of the involved targets, these GO annotations can associate involved targets with the GO terms that describe different functions of gene products. We then retrieve valid ancestral GO terms of the GO terms found by the GO annotations. A hierarchical GO network can thus be constructed based on the collected GO terms, in which the targets and GO terms are nodes, curated relationships between GO terms as well as between GO terms and targets (i.e., GO associations) are edges. This network describes the molecular function, cellular component, and biological process of the genes that encode the involved targets (Ashburner et al., 2000). The protein-protein interaction (PPI) network of the involved targets is extracted from DTINet. These two networks are termed as the GO-PPI network together.

To encode this GO-PPI network for generating hyperbolic target embeddings, we propose a fully Lorentz graph GNN-based target encoder. Specifically, the overall Gromovs hyperbolicity δ of the GO-PPI network is 1.5, theoretically, it could also be captured by hyperbolic models. Like the Euclidean spatial GNN, the essence of Lorentz GNN is aggregating the neighboring nodes' features or coordinates for the central node. The aggregation method used here is the Lorentzian centroid (Zhang et al., 2021; Yang et al., 2022), which is fully defined on Lorentz space and can prevent the aggregation deviating out of the Lorentz space. The aggregation is defined as follows:

$$AGG(x_i^{agg}) := \frac{\sum_{j \in N_i} a_{ij} x_j^{agg}}{\sqrt{-c}\, |\| \sum_{j \in N_i} a_{ij} x_j^{agg} \|_{\mathcal{L}}|} \tag{A.1}$$

where N_i is the neighboring node set of node i (including the self-loop), x_j^{agg} is the input Lorentz coordinate of node j in the aggregation, and a_{ij} is the aggregation weight allocated to the neighbor j of node i. For simplicity, we use the averaging aggregation weight similar to the vanilla GNN (Kipf and Welling, 2016). Based on this, a complete fully Lorentz GNN layer can be expressed as $AGG(FLLinear_{m,n}(x_i^{lin}))$, where x_i^{lin} is the input Lorentz coordinate of node i in its enhancement of the node feature (the enhancement is based on $FLLinear_{m,n}(x)$).

For the GO-PPI network, GO and PPI networks are aggregated by independent fully Lorentz GNN layers, to obtain hyperbolic target embeddings under GO and PPI respectively. To fuse these two groups of target embeddings, because there are few clear definitions of the feature fusion fully operated on Lorentz space, we use our defined fully Lorentz GNN layer to aggregate every target embedding under GO with corresponding embedding under PPI, to generate the final hyperbolic target embeddings.

A.3 Statistics of the independent test sets

Table A.1 and table A.2 provide relevant statistics of the independent test sets described in section 3.3.

TABLE A.1: The statistics of each independent repeat/test set in the comparison experiments.

DTINet	Repeat 1	Repeat 2	Repeat 3	Repeat 4	Repeat 5
Unseen drug number	48	48	56	62	52
Unseen target number	41	53	59	53	51
BioKG	Repeat 1	Repeat 2	Repeat 3	Repeat 4	Repeat 5
Unseen drug number	138	148	130	151	149
Unseen target number	75	78	79	69	83

TABLE A.2: The statistics of sample numbers of the different testing parts in extra ablation study.

	Repeat 1	Repeat 2	Repeat 3	Repeat 4	Repeat 5
Part 1 sample number	3120	5533	7558	6580	5609
Part 2 sample number	3621	4658	5163	4911	4862
Part 3 sample number	26307	25040	24569	24477	24562

A.4 Detailed experimental setup of NeoDTI and KGE_RF

Based on the same original drug-disease-target heterogeneous network extracted from the DTINet dataset and same sample data splitting, we added two representative network-based target prediction methods NeoDTI (Wan et al., 2019) and KGE_RF (Ye et al., 2021) into the performance comparison, which were trained on a standard 1:1 sampling of positive and negative samples. As for their testing, they were evaluated in the defined ranking task using original FLONE triplet test set (evaluation metrics: MRR etc.), due to that the above methods cannot effectively assign different target probability/ranking scores for the given drug under different diseases, therefore, for each testing drug-disease-target sample, their predicted probability score between the corresponding drug-target pair of the testing triplet was used as the final ranking score.

A.5 Fusing domain knowledge for the Euclidean-based model

On top of the conclusion obtained from section 3.3.4, to investigate the performance gain of Euclidean (translation-based) models from importing domain knowledge, based on the best Euclidean model FEC-FLONE$_{base}$ (measured by the extracted

TABLE A.3: Comparison results of the involved Euclidean models under 16 and 128 embedding dimensions based on the DTINet dataset.

Methods (dim=16)	MRR	Hits@1	Hits@3	Hits@10
FEC-FLONE$_{ECFP-SEQ}$	**0.4537**	**0.3842**	**0.4781**	**0.5958**
FEC-FLONE$_{ECFP-None}$	0.4312	0.3694	0.4482	0.5558
FEC-FLONE$_{None-SEQ}$	0.4145	0.3630	0.4295	0.5161
FEC-FLONE$_{base}$	0.3608	0.3174	0.3680	0.4484
Methods (dim=128)	MRR	Hits@1	Hits@3	Hits@10
FEC-FLONE$_{ECFP-SEQ}$	**0.4759**	**0.4113**	**0.5006**	**0.6025**
FEC-FLONE$_{ECFP-None}$	0.4602	0.4004	0.4815	0.5775
FEC-FLONE$_{None-SEQ}$	0.4519	0.4052	0.4627	0.5452
FEC-FLONE$_{base}$	0.4098	0.3705	0.4139	0.4907

[1]

DTINet DDT network and MRR), we did the following experiments, still keeping the same model evaluation settings with the previous sections.

Based on the better *ECFP*-based drug and *SEQ*-based target encoders (see section 3.3.4), we run the models FEC-FLONE$_{ECFP-SEQ}$, FEC-FLONE$_{ECFP-None}$, and FEC-FLONE$_{None-SEQ}$ on the DTINet DDT network, in which all hyperbolic operations were replaced by corresponding Euclidean ones, and the meaning of subscripts of these model names was consistent with that in section 3.3.4. As shown in table A.3, FEC-FLONE$_{base}$ obviously benefited from fusing both ECFP6 and target sequence similarity (i.e., FEC-FLONE$_{ECFP-SEQ}$). Compared with FEC-FLONE$_{base}$, based on MRR, 25.7% and 16.1% performance improvements were achieved on the dimensions 16 and 128, respectively. However, FLONE$_{ECFP-SEQ}$ still outperformed FEC-FLONE$_{ECFP-SEQ}$, which can further demonstrate the effectiveness of introducing Lorentz space embedding.

A.6 The advantages of the triplet-wise association prediction

We did an experiment to further illustrate the advantages of explicitly considering disease types when inferring targets of the given drug. As illustrated in section 3.1,

[1]The bold data indicates the best result under current evaluation metric.

the target and the corresponding given drug could be associated with specific diseases. In other words, given a drug and different diseases, the most associated targets could be different (due to the change of disease types). Formulating this to our ranking task: the best ranking targets of current drug could be different when considering different diseases. For the binary DTI association prediction method, it predicts the pairwise association between drugs and targets, i.e., it will assign the same probability score to a target of the given drug no matter what the associated diseases are. Therefore, explicitly considering the disease type to formulate current target identification task as a triplet-wise association ranking/prediction task could bring finer scale virtual screening, allowing models to identify drug targets under particular disease types.

We devised an experiment to further demonstrate this, take drug *DB00363* (*Clozapine*) and target *P35348* (*ADRA1A*) as an example, for our DDT network screened from the DTINet dataset, only 265 out of 1160 diseases were recorded to have associations with above DTI pair. Theoretically, the model should assign higher ranking to *P35348* compared with other potential targets of *DB00363* under these 265 diseases, and assign the relatively lower ranking (to *P35348*) under the rest of the diseases (although these diseases may include sparse unidentified positive diseases associated with *DB00363-P35348* pair), instead of assigning the total equal ranking under all diseases. Based on this, we extracted all 1160 triplets corresponding to *DB00363*, *P35348*, and the 1160 diseases as the test set. The way of splitting training and validation sets was same to the setting in section 3.3.2 (random 6:2 DTI pair-based triplet splitting). We recorded the evaluation results of FLONE (FLONE$_{ECFP-SEQ}$) for the test triplets corresponding to the positive diseases and negative diseases separately (table A.4). We found that FLONE did effectively allocate higher ranking for *P35348* under the positive diseases (MRR: 0.9504) and *DB00363*, and allocate low ranking for *P35348* under the negative diseases and *DB00363* (MRR: 0.0964), which further demonstrated the advantages of the triplet-wise association predictions.

TABLE A.4: The evaluation results on the test drug-disease-target triplets corresponding to positive diseases and negative diseases (on DTINet).

	Hits@10	Hits@3	Hits@1	MRR
Positive diseases	0.9736	0.9547	0.9396	0.9504
Negative diseases	0.2670	0.0603	0.0145	0.0964

A.7 The projection details of Lorentz and Euclidean embeddings

For projecting the high-dimensional Lorentz embeddings to 2-dimensonal (2D), first these Lorentz embeddings are mapped to their isomorphic Poincaré embeddings using equation (A.2), where P^i is the i_{th} Poincaré embedding, and L_s^i and L_t^i are the spatial and time dimensions of corresponding Lorentz embedding L^i:

$$P^i = \frac{L_s^i}{L_t^i + 1} \tag{A.2}$$

Then, to project all the high-dimensional Poincaré and Euclidean target embeddings to the corresponding 2D space for visualization, below equation (A.3) and equation (A.4) by (Balazevic, Allen, and Hospedales, 2019) were used, these formulas can keep the original distances and angles of these target embeddings relative to the given drug embedding:

$$E_i^x = \frac{E^D}{\parallel E^D \parallel} E_i, i \in \{D, T_0, T_1, ..., T_N\} \tag{A.3}$$

$$E_i^y = \sqrt{\parallel E_i \parallel^2 - \parallel E_i^x \parallel^2}, i \in \{D, T_0, T_1, ..., T_N\} \tag{A.4}$$

where E_i^x and E_i^y are the coordinate on the x-axis and y-axis of i_{th} new projected embedding, E^D is the original high-dimensional embedding of the given drug, N is the number of candidate target entities, and $\parallel \cdot \parallel$ represents the Euclidean norm. After that, the above Lorentz target embeddings have been mapped into a 2D Poincaré disk in which the hyperbolic embedding quality can be effectively checked.

Appendix B

Supplementary for Chapter 4

B.1 The illustration of steps of collecting our dataset

As illustrated in figure B.1, to retrieve relevant data for a drug-drug-cell line pair (combination), for example, $D1 - D2 - C1$, we first make this sample equipped with a synergy score and 20 adverse effect labels (with at least one known/positive value). Next, retrieve ECFP6 corresponding to $D1$ and $D2$ to characterize their chemical structural property. Next, obtain targets of $D1$ and $D2$ from the drug-target set (based on the DrugComb dataset, TWOSIDES dataset, and Luo et al. dataset). Then, from the PPI set (based on the TWOSIDES dataset), obtain protein-protein interaction (PPI) information of the targets retrieved in the last step. Finally, from the DepMap database, retrieve expression data of genes that belong to the targets in the drug-target set. All this information will be utilized by Muthene for predicting the synergy score of $D1 - D2 - C1$.

B.2 Threshold selection for drug-drug TE relationships

For drug-drug TE relationships, we give an effective edge between drug i and drug j based on a threshold originated from corresponding therapeutic synergy scores (Peng et al., 2021), the detailed steps are as follows.

we first calculate the value at the 95% percentile of the standard normal distribution, which is approximately 1.64. Then, we calculate the overall mean value (μ) and variance (var) of synergy score samples. Next, for each synergy score s_{ijk} that represents the therapeutically synergistic effect degree between drug node i, drug node j, and cell line k pair, we calculate its $z - score = \frac{(s_{ijk} - \mu)}{\sqrt{var}}$. If any of $z - score$ between

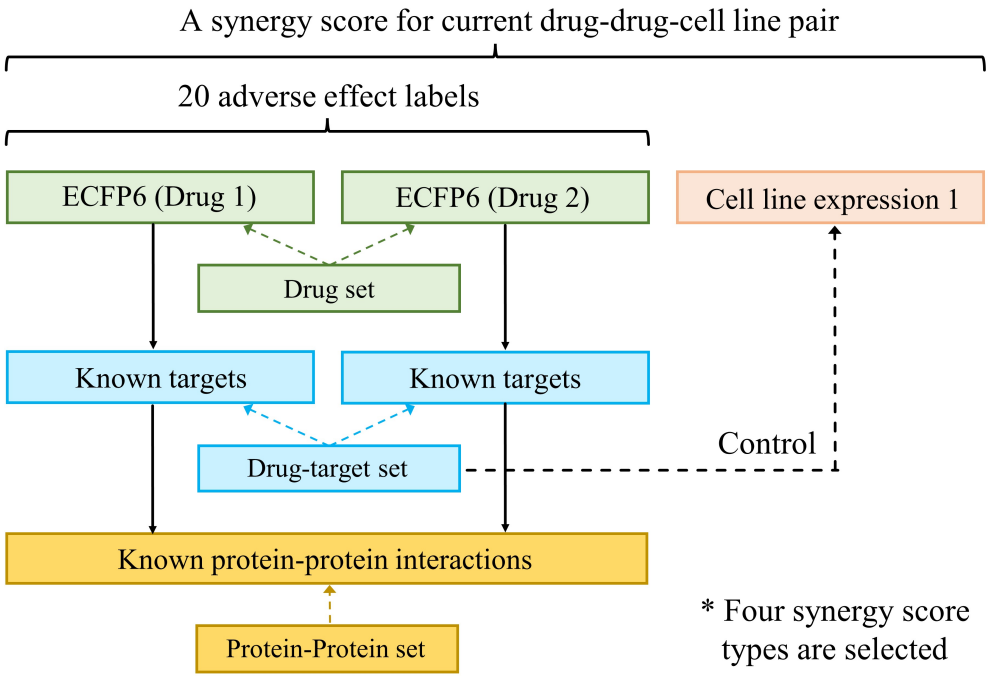


FIGURE B.1: The illustration of collecting relevant data for a drug-drug-cell line pair.

drug node i and drug node j is larger than or equal to 1.64, the drug i-drug j TE relationship value is set to 1, indicating that there exists an effective edge, otherwise it is 0.

B.3 Model running environment and hyper-parameter settings

To demonstrate our hypothesis, we use Python 3.6.13 and Pytorch 1.10.2 to build the framework of involved models, all models are run on the same server with the configuration of Linux 4.18, CUDA tool kit 11.3.1, Intel Xeon Platinum 8360Y, and NVIDIA A100-SXM. The basic hyper-parameters of Muthene, DeepSynergy, and their variants are shown in table B.1 and table B.2. For DeepDDS and TranSynergy, we adopt their recommended basic hyper-parameters in their original papers.

TABLE B.1: The hyper-parameters shared by all Muthene variants in the comparison experiment.

Hyper-parameter names	Hyper-parameter values
Meta-path embedding dimension	64
Weighted aggregation times K in GAT	8
Cell line feature dimension d^{cell}	64
Neuron numbers of conic layers in DNN_{TE}	2048, 1024, 512

TABLE B.2: The hyper-parameters shared by all DeepSynergy variants in the comparison experiment.

Hyper-parameter names	Hyper-parameter values
Neuron numbers of conic layers	2048, 1024, 512

B.4 Discussion of involved three representative methods

For the mentioned three representative methods in the comparison, DeepSynergy achieved overall higher prediction accuracy compared with DeepDDS and TranSynergy. We thought the main reason was that TranSynergy requires the drug target interaction information as drug features, in which involved targets/genes should be consistent with the genes included in cell line features. The performance of TranSynergy would be limited when the lack of interactive information of above specific target/genes for involved drugs (Liu and Xie, 2021). While in our dataset, 11 out of 106 drugs do not include the interactions with these targets/genes, leading to the sub-optimal results of the method. For DeepDDS, its drug features are generated by GNNs (from learning drug molecular graphs). However, we used a stricter model evaluation setting (i.e., drug pairs in the test and validation sets will not occur in the training set), in this case, GNNs cannot produce effective drug features for the drugs that have not seen in the training set (e.g., in one fold of the independent repeats, 14/16 types of drugs were in the validation/test set but not in the training set), making it difficult to be generalized to drug-drug-cell line pairs corresponding to these drugs.

B.5 Detailed description of Mut-GIN

Drug chemical feature information is important for drug-drug combination related predictions (Huang et al., 2020). To test the effectiveness of using the learnt drug chemical feature embeddings to replace commonly used ECFP6 features, we design a GNN-based module to produce the new embeddings. Specifically, for each drug node, we acquire its molecular graph with atoms as nodes, chemical bonds as edges, and numerical encoding of the atomic number as the initial node embedding. The Graph Isomorphism Network (GIN) is adopted to learn each molecular graph, as it has close distinguishing capacity to the Weisfeiler-Lehman (WL) test in learning graph structures and has already been demonstrated to have better performance in many bioinformatics graph classification tasks compared with the vanilla GNN (Xu et al., 2018).

Formally, for a molecular graph of a drug, the embedding of a atom node i at layer $l+1$ is updated as follows:

$$h_i^{(l+1)} = \sigma(MLP^{l+1}((1+\epsilon)h_i^{(l)} + sum(\{h_j^l, j \in N(i)\}))) \tag{B.1}$$

where ϵ is a trainable parameter and $N(i)$ is the neighbor set of atom node i. After stacking multiple GIN layers, all atom embeddings of a drug are aggregated through the global mean pooling (i.e., *readout*) function, to generate the chemical feature embedding for the compound c:

$$z_c^G = readout(\{h_i^{(l+1)} | i \in G\}) \tag{B.2}$$

This module will be trained along with the main modules of Muthene in an end-to-end way for joint optimization (denoted as Mut-GIN).

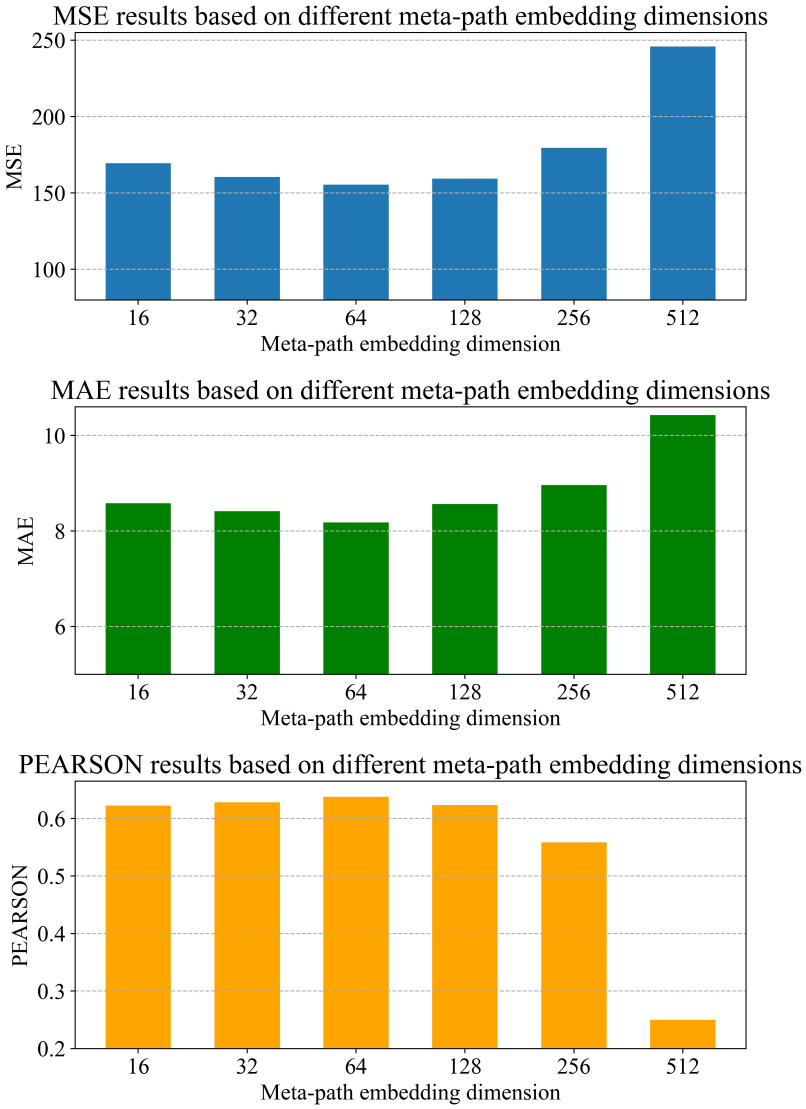


FIGURE B.2: Different model evaluation results given various meta-path embedding dimensions.

B.6 Ablation study for dimension of meta-path embeddings

In order to investigate the performance influence caused by different choices of meta-path embedding dimensions, based on the Loewe synergy score, we report the performance change of Muthene when changing the meta-path embedding dimension while fixing other hyper-parameters (during one of the five times data splitting), and the results are shown in figure B.2.

From the results, we observe that, MSE, MAE, and Pearson gradually become

better with the increase of meta-path embedding dimension, and then the performance gets worse after reaching a threshold, and 64 selected in our comparison experiments achieves the best results among these dimension choices.

Appendix C

Supplementary for Chapter 5

C.1 Pre-training dataset collection procedure

To retrieve enough real protein-protein complexes for the pre-training, we first collected all raw protein-protein complex Protein Data Bank (PDB) files from the PDBbind (PDBbind v2020 protein-protein complexes) (Su et al., 2018) and 3DComplex (QS40 level) (Levy et al., 2006) databases. In order to prevent implicit information leakage, based on their PDB ID, we removed the complexes which share either high topological structural similarity or sequence identity with the complexes in downstream datasets from above two datasets. This is done by 1) ignoring the complexes included in all downstream datasets, 2) ignoring incomplete complexes (i.e., the complexes with only $C\alpha$ in backbone atoms) and over-large complexes (i.e., the complexes with residue number over 3000), and 3) performing further complex screening based on the Evolutionary Classification of protein Domains (ECOD) classification system (Cheng et al., 2014).

For the ECOD-based complex screening, specifically, ECOD has clustered a large number of commonly-seen protein-protein complexes based on their protein domains under different pre-defined scales, the complexes clustered into the same group (under a certain pre-defined scale) represent that their included domains share high similarity under the current scale. For example, domains in the same T-scale group are the domains sharing similar topological connections, F-group groups domains with significant sequence similarity in a family. Based on this, we first downloaded the off-line ECOD classification file (ecod.develop287.domains.txt), and retrieved the T-group and F-group category sets based on protein domains

included in every wide-type protein-protein complex of all downstream datasets. Next, for any complex in PDBbind or 3DComplex, it was selected as a pre-training sample only if there was no overlap between the allocated effective T-group/F-group categories of its any domains and the above T-group/F-group category sets. After that, 14936 complexes were collected in total, all of which share low topological and sequence similarity with all downstream data.

C.2 The generation of the mutant protein-protein complex structure

In the downstream $\Delta\Delta G$ prediction task, we considered three types of mainstream molecular generation tools for providing mutant (MT) protein-protein complex structures for performance comparison, including FoldX, MODELLER, and AlphaFold2.

Specifically, for FoldX, we used the BuildModel function in FoldX 4.0, inputting the wild-type (WT) structure and corresponding mutation information (mutation sites and AA mutation types), to produce the mutant structure.

For MODELLER, the structure for a single mutation or multiple point mutations was modeled based on its WT structure using homologous modeling with MODELLER 10.4. We added chain break (‘/’) characters to the alignment file. Then we used the AutoModel class to build models of the complex.

For AlphaFold2, we used a locally installed version of the publicly available ColabFold4 to predict mutant structures. The amino acid sequence was modified to reflect the desired mutations, and this updated sequence was used as input for the AlphaFold2. Following the AlphaFold2 pipeline, we performed a MSA search with MMseqs2 on the UniRef30 database5 and generated five models for each sample. These models were subsequently relaxed with the Amber program and ranked. We ultimately selected the top-ranked structure for $\Delta\Delta G$ prediction tasks.

C.3 Baseline Methods

We compare our MpbPPI against seven baseline methods in total, including DGCddG, PerSpect-E, GeoPPI, TopNetTree, FoldX, Flex ddG, and MM/GBSA. We briefly describe these methods as follows.

- DGCddG (Jiang et al., 2023): DGCddG is a graph neural network (GNN)-based method, mainly designed for estimating PPI $\Delta\Delta G$ caused by amino acid (AA) mutations occurring at PPI interfaces. It creates residue-level contact graphs for WT and MT protein-protein complexes, in which edges in the WT graph are constructed based on the distance threshold between residues and the mutant graph is created by simply replacing WT AA nodes to MT AA nodes. Besides, it uses biochemical (e.g., polarity) and evolution information (e.g., PSI-Blast profile) as node features. The final PPI $\Delta\Delta G$ is predicted based on the WT and MT representations from a multi-layer GNN and extra DSSP (Kabsch and Sander, 1983) information.

- PerSpect-EL (Wee and Xia, 2022): PerSpect-EL is designed to use persistent spectral-based features that capture molecular-level information of PPI systems represented by a series of simplicial complexes. The predictions are made by the stack of three different types of base learners.

- GeoPPI (Liu et al., 2021): GeoPPI is a GNN-based method for capturing atom-level relationships, different from DGCddG, its construction of WT and MT complex graphs is based on atom distance relationships. Specifically, GeoPPI creates atom-level graphs for the WT and MT structures, in which an edge is added if the distance between two atoms is smaller than 3Å and an extra cutoff is set to only retain atoms around mutation or interface sites to save the computational cost. To better capture the atom-level relationships in complex graphs, a single-task perturbation-based pre-training task is utilized to train the GNN model.

- TopNetTree (Wang, Cang, and Wei, 2020): TopNetTree aims to capture inter-atom geometric relationships of protein-protein complexes based on

an algebraic topology techniques named persistent homology (Edelsbrunner, Letscher, and Zomorodian, 2000), to generate the representation for the protein-protein complex considering mutation information. Furthermore, the generated PPI representation is further enhanced by a convolutional neural network (CNN) and sent to a gradient-boosting trees (GBT) model for PPI $\Delta\Delta G$ prediction.

- Flex ddG (Barlow et al., 2018): Rosetta Flex ddG is the latest protein affinity maturation module released in 2018. It uses "backrub" to create a set of models after sampling the conformation, minimizing torsion, repacking the side chain, and average to the set method to predict the interface binding affinity.

- MM/GBSA (Wang et al., 2019a; Salomon-Ferrer, Case, and Walker, 2013): Molecular Mechanics Generalized Born Surface Area (MM/GBSA) is a free energy prediction method developed by Kollman et al. and balances the computational accuracy and efficiency between the docking score function method and the alchemical free energy method. In this study, the proteins were originally mutated using PyMOL (DeLano et al., 2002). The protein-protein complex systems were constructed using the “tleap” program in Amber22, and the AMBER ff14SB force field was optimized for the protein. In order to avoid unexpected local maxima, 500 steps of minimizations were carried out for each protein-protein complex using the steepest descent and conjugate gradient method, where the cutoff point for non-bonded interactions was set to 12Å. The “sander” program in Amber22 was utilized to carry out the minimizations. We perform MM/GBSA calculations based on the results of the energy minimization of the complex system using the MMPBSA.py module in AMBER22.

TABLE C.1: The RMSE and MAE results of MpbPPI with different mutant structure sources under the mutation-level tenfold cross-validation on the four PPI $\Delta\Delta G$ datasets. The results of MpbPPI were reported based on the average of five times of the independent tenfold cross-validation runs.

S645	RMSE	MAE
MpbPPI-FoldX	**1.537±0.018**	**0.971±0.010**
MpbPPI-MODELLER	1.562±0.010	0.996±0.003
MpbPPI-AlphaFold2	1.590±0.022	1.018±0.011
M1101	RMSE	MAE
MpbPPI-FoldX	**1.317±0.015**	**0.839±0.008**
MpbPPI-MODELLER	1.338±0.013	0.861±0.008
MpbPPI-AlphaFold2	1.372±0.020	0.881±0.008
S1131	RMSE	MAE
MpbPPI-FoldX	**1.250±0.004**	**0.878±0.004**
MpbPPI-MODELLER	1.328±0.004	0.966±0.003
S4169	RMSE	MAE
MpbPPI-FoldX	**1.073±0.004**	**0.717±0.003**
MpbPPI-MODELLER	1.111±0.008	0.754±0.004

[1]

C.4 The RMSE and MAE results of MpbPPI with different mutant structure sources under the mutation-level tenfold cross-validation

The RMSE and MAE results of MpbPPI with different mutant structure sources under the mutation-level tenfold cross-validation (on the four PPI $\Delta\Delta G$ datasets) are provided in table C.1.

C.5 Further result providing and analysis of MpbPPI with different mutant structure sources

The MpbPPI with different mutant structure sources-related further result providing and analysis are displayed in figure C.1 and figure C.2.

[1]The bold data indicates the best result under current evaluation metric and dataset.

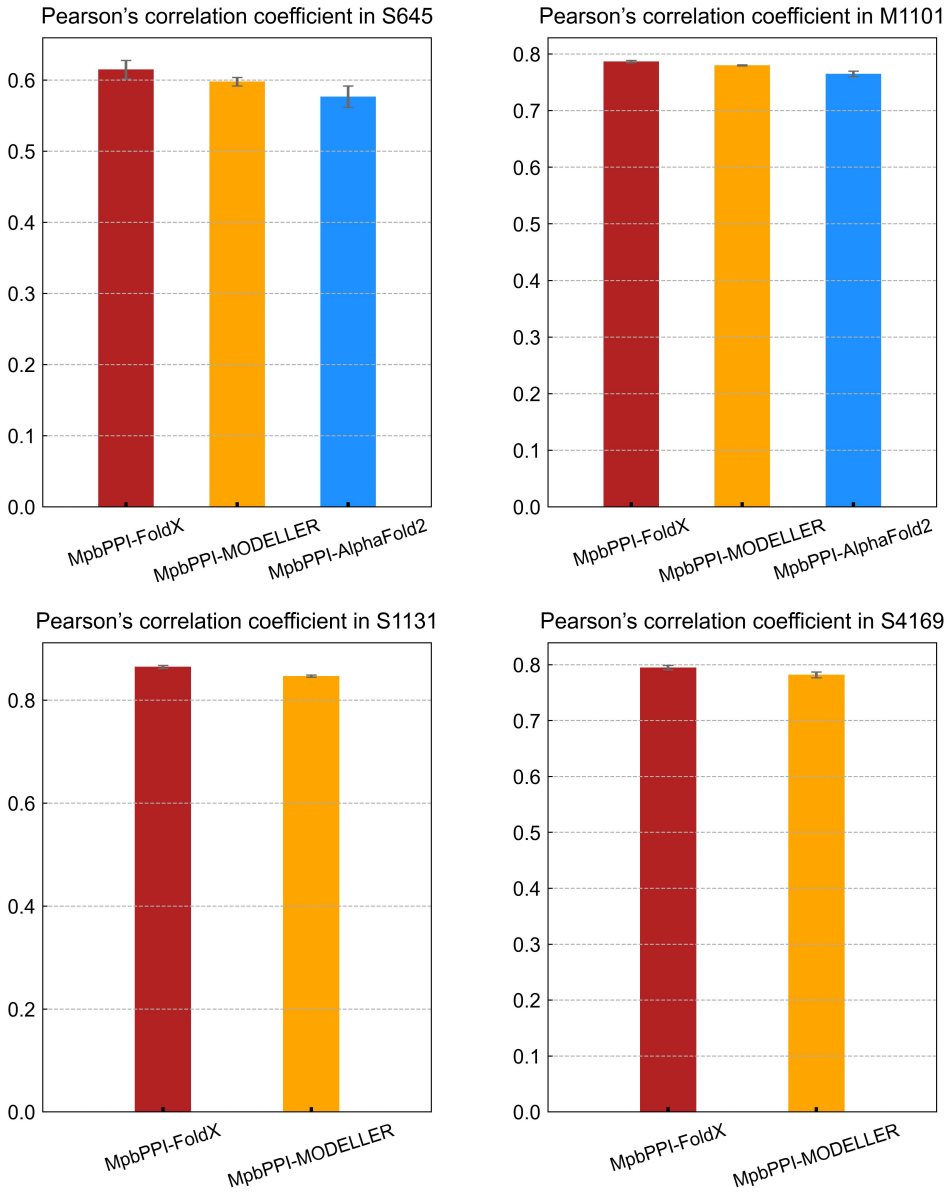


FIGURE C.1: The R_P results of involved MpbPPI variants under the mutation-level tenfold cross-validation on the four PPI $\Delta\Delta G$ datasets.

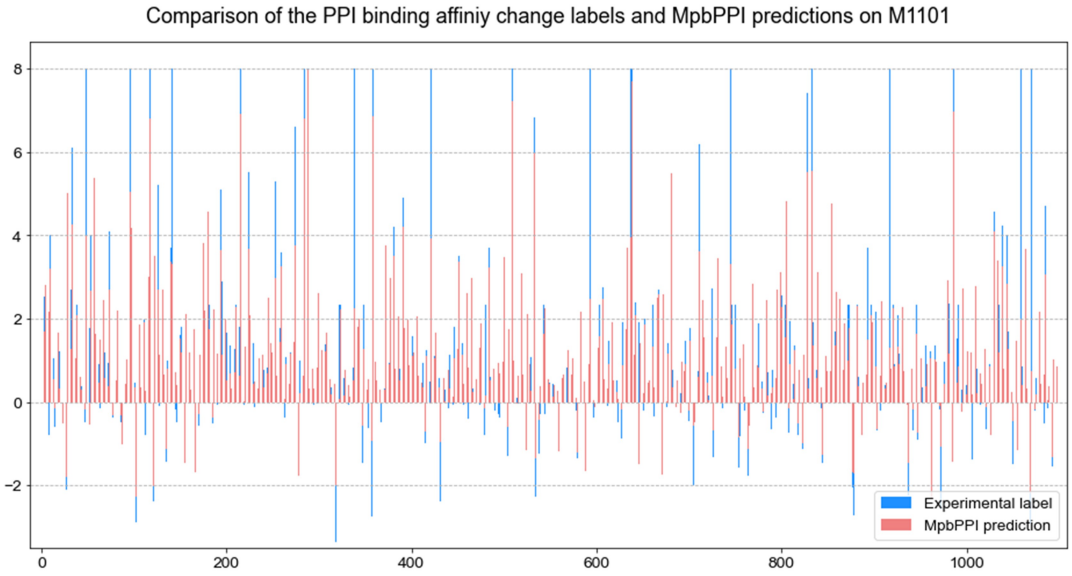


FIGURE C.2: The test set prediction results of MpbPPI-FoldX under the mutation-level ten-fold cross-validation on M1101 (unit: kcal·mol-1). The red bars are the model predictions while the blue ones are corresponding experimental PPI $\Delta\Delta G$ ground truths. Taking the predictions on M1101 as an example, we observed that, MpbPPI-FoldX can successfully give most of the predictions a correct positive or negative sign, i.e., correct binding affinity increase or decrease. In addition, for the outlier samples (with $\Delta\Delta G$ equaling to 8 kcal·mol-1), the prediction accuracy is relatively low and consequently influences the overall accuracy. We thought a contributing reason is that such outliers are different from other ground truths and the sample number of them is relatively small, leading to the difficulty of model fitting.

C.6 Experiment about correlating sequence representation difference to the PPI $\Delta\Delta G$ value

To investigate the results of directly correlating difference (e.g. distance, similarity) between sequence representations of the mutant and wildtype complexes, to the PPI $\Delta\Delta G$ value, we did an experiment as follows:

Under the largest dataset S4169, for each sample point, we first retrieved all initial residue node features corresponding to the WT and mutant protein-protein complexes separately (i.e., these are the residue features before being sent to the trained GEE encoder and GBT-based decoder). Then, to generate the (sequence) representation of the WT sequence, we concatenated the corresponding residue node features following the order of the WT residue sequence. As an analogy, the representation of the mutant sequence can also be generated (these two representations have the same length for the same sample point). Next, we calculated the cosine similarity (which

TABLE C.2: R_P results of the re-produced representative methods.

Name	Dataset	R_P
TopNetTree	S645	0.648±0.007
TopNetTree	S1131	0.857±0.003
GeoPPI	S4169	0.719±0.006

are commonly used to compare the difference of representations) between the mutant and WT sequence representation for each sample point, and used Pearson's correlation coefficient (R_P) to correlate the calculated cosine similarity to the PPI $\Delta\Delta G$ ground truth. The R_P on S4169 was 0.255 while our tenfold cross-validation result was 0.795 (based on FoldX), which further demonstrated the necessity of training a predictive model on top of these representations.

C.7 Experiment about reproducing results of some representative baseline methods based on available information

We re-produced the results of GeoPPI on the S4169 dataset and TopNetTree on the S645 and S1131 datasets. Specifically, GeoPPI is a representative geometric property pre-training-based method and TopNetTree is also an important topology-based machine learning method for PPI $\Delta\Delta G$ predictions. Furthermore, GeoPPI provides the pre-trained feature encoder/embedder and exact hyper-parameters of its decoder on the S4169 dataset and TopNetTree provides the already pre-processed sample point features (and corresponding decoder hyper-parameters) on the S645 and S1131 datasets, on which we can apply the identical tenfold cross-validation settings for comparing them (table C.2).

From the results, we observed that, MpbPPI still outperformed TopNetTree and GeoPPI on relatively larger datasets S1131 (MpbPPI-FoldX: 0.865±0.003) and S4169 (MpbPPI-FoldX: 0.795±0.004), respectively [except for TopNetTree on S645 (MpbPPI-FoldX: 0.615±0.013)], which further demonstrated the effectiveness of our proposed framework.

TABLE C.3: The RMSE and MAE results of involved MpbPPI variants under the mutation-level tenfold cross-validation. The results of the MpbPPI variants were reported based on the average of five times of the independent tenfold cross-validation runs.

S645	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**1.537±0.018**	**0.971±0.010**
MpbPPI (Backb+SASA+AA)	1.601±0.015	0.994±0.008
MpbPPI (Backb+Sidec+AA)	1.635±0.010	1.058±0.005
MpbPPI (Backb+AA)	1.715±0.018	1.053±0.012
MpbPPI (w/o AA)	1.540±0.010	1.058±0.005
M1101	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**1.317±0.015**	**0.839±0.008**
MpbPPI (Backb+SASA+AA)	1.387±0.013	0.860±0.007
MpbPPI (Backb+Sidec+AA)	1.394±0.017	0.891±0.005
MpbPPI (Backb+AA)	1.447±0.020	0.885±0.006
MpbPPI (w/o AA)	1.325±0.012	0.835±0.009
S1131	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	1.298±0.006	0.943±0.002
MpbPPI (Backb+SASA+AA)	**1.250±0.004**	**0.878±0.004**
MpbPPI (Backb+Sidec+AA)	1.311±0.003	0.948±0.004
MpbPPI (Backb+AA)	1.325±0.005	0.936±0.006
MpbPPI (w/o AA)	1.372±0.010	0.963±0.006
S4169	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	1.088±0.004	0.737±0.001
MpbPPI (Backb+SASA+AA)	**1.073±0.004**	**0.717±0.003**
MpbPPI (Backb+Sidec+AA)	1.120±0.005	0.762±0.002
MpbPPI (Backb+AA)	1.130±0.005	0.755±0.003
MpbPPI (w/o AA)	1.133±0.007	0.752±0.004

[2]

C.8 The RMSE and MAE results of MpbPPI with different pre-training task combinations under the mutation-level tenfold cross-validation

The RMSE and MAE results of MpbPPI with different pre-training task combinations under the mutation-level tenfold cross-validation (on the four PPI $\Delta\Delta G$ datasets) are provided in table C.3.

[2]The bold data indicates the best result under current evaluation metric and dataset.

C.9 Detailed description of the protein-protein complex type-based data splitting

The specific data splitting pipeline can be summarized as follows:

1. For a downstream PPI $\Delta\Delta G$ dataset, we first obtain its included WT protein-protein complex types, and count the sample point number for each complex type. Then, sort the complex types based on their corresponding sample point numbers from large to small.

2. For the sorted WT complex types, we iterate all of them (from large to small based on their sample point numbers) and put every five in one bin, and the last bin could be incomplete as the total number of WT complex types may not be a multiple of 5.

3. For all bins except for the last one, the WT complex types within a bin are randomly shuffled and then are allocated into different five folds (one WT complex type only to one fold). After that, for each fold, the sample points corresponding to its allocated WT complex types are added into this fold. Next, the sample point number contained in each fold is counted.

4. For the last bin, its WT complex types (and the corresponding sample points) are preferentially allocated into folds that have fewer existing sample points (one WT complex type only to one fold), which makes the sample point number and WT protein-protein complex type number assigned to each fold to be close.

C.10 The illustrative example of the predicted outcomes of MpbPPI

The H-Ras/Raf complex is an important component of the cellular signaling pathway, known as the Ras/Raf/MEK/ERK pathway (also called the mitogen-activated protein kinase (MAPK) pathway). This pathway plays a critical role in regulating cell growth, differentiation, survival, and apoptosis. MpbPPI predicted $\Delta\Delta G$ values

of 1.44 kcal/mol and 1.61 kcal/mol for H-Ras protein mutations T35A and E37G in the H-Ras/Raf complex, respectively. These predictions suggested that the two mutations decreased the interaction between H-Ras and Raf, which was consistent with previous experimental finding that these mutations hinder the interaction between H-Ras and Raf (Drugan et al., 1996). Moreover, antibodies are proteins employed by the immune system to identify and neutralize foreign particles, including viruses and bacteria. Gaining insights into how mutations influence protein-protein interactions can help researchers develop antibodies that exhibit enhanced binding to target proteins. Our model, MpbPPI, has the potential to aid researchers in creating superior antibodies (e.g., help to evaluate $\Delta\Delta G$ between newly designed antibodies with AA alterations and target proteins compared with the original protein-protein interaction systems), ultimately resulting in improved therapeutic approaches and better outcomes for patients.

C.11 The RMSE and MAE results of MpbPPI with different pre-training task combinations under the wild-type protein-protein complex type-based cross-validation

The RMSE and MAE results of MpbPPI with different pre-training task combinations under the wild-type protein-protein complex type-based cross-validation (on the four PPI $\Delta\Delta G$ datasets) are provided in table C.4.

C.12 The discussion of some optional features for $\Delta\Delta G$ predictions

1. Electrostatic energy information: The interaction between protein complexes is mainly non-bonded interactions, particularly electrostatic interactions. In some previous machine learning or deep learning prediction models, the PPI contact interface electrostatic energy was used as a feature to more accurately predict the impact of amino acid mutations. Electrostatic energy is the energy

[3]The bold data indicates the best result under current evaluation metric and dataset.

TABLE C.4: The RMSE and MAE results of involved methods under the cross-validation that splits data based on the wild-type protein-protein complex types.

S645	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**1.765±0.037**	**1.180±0.021**
MpbPPI (Backb+SASA+AA)	1.792±0.033	1.186±0.019
GeoPPI	-	-
FoldX	2.637	1.600
Flex ddG	3.842	2.250
MM/GBSA	8.102	7.174
M1101	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**1.931±0.042**	**1.367±0.025**
MpbPPI (Backb+SASA+AA)	1.973±0.046	1.376±0.036
GeoPPI	-	-
FoldX	3.429	2.278
Flex ddG	4.494	2.875
MM/GBSA	5.925	4.496
S1131	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**2.450±0.080**	**1.772±0.073**
MpbPPI (Backb+SASA+AA)	2.586±0.063	1.808±0.059
GeoPPI	-	-
FoldX	2.661	1.776
Flex ddG	3.505	2.346
MM/GBSA	6.518	5.076
S4169	RMSE	MAE
MpbPPI (Backb+Sidec+SASA+AA)	**1.636±0.009**	**1.068±0.017**
MpbPPI (Backb+SASA+AA)	1.682±0.017	1.086±0.019
GeoPPI	1.899±0.009	1.272±0.019
FoldX	2.263	1.461
Flex ddG	4.488	2.972
MM/GBSA	4.962	3.909

resulting from the interaction between charges and can be obtained by calculating the Coulomb interaction energy between molecules in the molecular force field. In this study, however, our features only included the sequence and geometric characteristics of the protein structure, without considering energy features related to electrostatic interactions. Notably, the mutant structures input into our model were obtained using FoldX, MODELLER, and AlphaFold, all of which perform energy optimization to obtain the structures. The molecular energy optimization process includes the calculation of electrostatic interaction energy. Therefore, although our features did not explicitly include electrostatic energy, electrostatic interactions may implicitly affect the accuracy of the model through the generated mutated structures.

On top of this, we also discussed the prediction results of charged residues mutations of MpbPPI here: The four test datasets in this study all contain charged amino acids. Specifically, in the largest dataset, S4169, the MpbPPI-FoldX predicted R_P values for charged-to-neutral and neutral-to-charged residue mutations were 0.807 and 0.861 (based on the tenfold cross-validation), respectively, which were higher than the overall R_P value (0.795) for the dataset. This indicated that our model had good performance in predicting charged residue mutations.

2. Entropic effect information: In PPI, entropy change is often associated with the free energy change that occurs when proteins transition from their unbound state to the bound state (Kamisetty et al., 2011). As proteins bind to one another, they generally undergo conformational changes, leading to a decrease in conformational entropy. However, calculating entropy is complex and computationally expensive, posing challenges for incorporating it into both classical computational models (such as molecular dynamics simulations) and machine learning models.

   Standard methods for estimating the entropic component include normal mode analysis (NMA) (Genheden et al., 2012) and the quasi-harmonic approximation (Chang, Chen, and Gilson, 2005). In machine learning, entropic effects may be implicitly incorporated into the feature vector using descriptors that

capture the protein's flexibility or mobility, such as the number of rotatable bonds or the radius of gyration (Rose, 2021).

In this study, we did not create feature vectors that explicitly account for entropy contributions. When evaluating the model's accuracy, it is crucial to consider whether entropy effects are expected to play a significant role in the systems being studied. We assessed the impact of AA mutations in the complex by predicting the relative binding free energy ($\Delta\Delta G$) between the mutant and WT structures. Since the structures before and after the mutation are relatively similar, their contributions to conformational entropy are also similar and can largely cancel each other out. This allows us to disregard the impact of the entropy term.

3. Multiple sequences alignment (MSA) information: Regarding MpbPPI and MSA, MpbPPI did not directly use MSA as input. Instead, it used simple geometric features of the protein complex structure, such as the unit vector of $C\alpha$-side chain, to efficiently capture the relationships between the backbone and side chains of the protein. Using these geometric features, MpbPPI was able to predict the effect of AA mutations on protein-protein interactions. However, in the combination of MpbPPI and AlphaFold2, the MSA was used by AlphaFold2 to identify conserved regions and patterns in the protein sequence, which could help guide the prediction of the mutant structure.

   In some cases, MSA can provide valuable information about the evolutionary relationships between related proteins and help identify functionally important residues and regions in protein sequences. However, there are some limitations about using MSA as input features, such as the potential for incomplete or biased MSA and the computational expense of generating them. Despite these challenges, MSA remains an effective tool for predicting protein-protein interactions, and many models continue to use them as input features. In the future work, we will attempt to incorporate more flexible biochemical and evolutionary information, which is compatible with the existing geometric features and pre-training tasks under current downstream task, into our MpbPPI framework.

C.13 Detailed description of the residue node feature calculation

We first denote all the preliminarily normalized backbone and side chain coordinate sets in current contact graph $\mathbb{G}$ (see section 5.3.1), which represent all backbone and side chain atom sets in $\mathbb{G}$, as $C^{B^{'}}$ and $C^{S^{'}}$ (the corresponding sets for an individual residue i are denoted as C_i^B and C_i^S, respectively). Based on this, the detailed description of devised seven types of node features f_i (mentioned in section 5.3.1) are as follows, in which the reason for the further normalization is to make all features basically stay in the same scale for the better model optimization:

1. The centroid of side chain atoms (vector): further normalize $C^{B^{'}}$ and $C^{S^{'}}$ based on the largest Euclidean distance between $C^{B^{'}}$ and the origin of coordinate system (i.e., (0, 0, 0)), to acquire $C^{B^{''}}$ and $C^{S^{''}}$, and then the centroid of $C_i^{S^{''}}$ is calculated as the residue feature.

2. The center of mass of side chain atoms (vector): based on the calculated $C_i^{S^{''}}$, to calculate the center of mass of $C_i^{S^{''}}$ as the residue feature (i.e., the centroid weighted by corresponding atom mass).

3. The maximum coordinate value of side chain atoms (vector): based on the above $C_i^{S^{''}}$, to retrieve the maximum coordinate values of $C_i^{S^{''}}$ on X-, Y-, Z- axis, respectively.

4. The unit vector of $C\alpha$-side chain geometric relationships (vector): denote the $C\alpha$ coordinate of residue i after preliminary centroid-based normalization as $C\alpha_i^{'}$, calculate the forward and reverse unit vectors in the directions of $C\alpha_i^{'} - C_i^{S^{'}}$ and $C_i^{S^{'}} - C\alpha_i^{'}$ as the residue feature.

5. The normalized $C\alpha$ coordinate (vector): the $C\alpha$ coordinate in $C_i^{B^{''}}$.

6. Solvent-accessible surface area (SASA) (scalar): calculate the SASA of residue i in current protein-protein complex based on the probe radius 1.4Å (denoted as $\text{SASA}_{complex_i}$. Such calculated SASA can depict the surface outline of the side chain atoms of a residue, and also suggest the geometric relationships of these atoms (Liu et al., 2022).

7. Residue interface information (scalar): PPI interface information has been demonstrated to be effective in PPI $\Delta\Delta G$ predictions (Liu et al., 2021). Therefore, based on above $\text{SASA}_{complex_i}$, further calculate the SASA of residue i in current chain (denoted as SASA_{chain_i}), if $\text{SASA}_{chain_i} - \text{SASA}_{complex_i}$ (i.e., dASA_i) is larger than 1Å^2, treat i as an interface residue and mark this feature as 1, otherwise mark it as 0.

C.14 Equations of the GVP-GNN backbone

Concisely, GVP-GNN uses the message passing strategy (Gilmer et al., 2017) for geometric neighboring learning to produce the (residue) node-level representations:

$$f_{(i,j)}^{message} = GVP(concat(f_j^{(S,V)}, f_{(i,j)}^{(S,V)})) \tag{C.1}$$

$$f_i^{(S,V)} = LayerNorm(f_i^{(S,V)} + mean(Dropout(\sum_{j:\, edges\, of\, j\, to\, i \in \mathcal{E}} f_{(i,j)}^{message}))) \tag{C.2}$$

$$f_i^{(S,V)} = Layernorm(f_i^{(S,V)} + Dropout(GVP(f_i^{(S,V)}))) \tag{C.3}$$

in which $f_j^{(S,V)}$ and $f_{(i,j)}^{(S,V)}$ are the tuples of f_j^S (i.e., residue node scalar features) and f_j^V (i.e., residue node vector features) and the tuples of $f_{(i,j)}^S$ (i.e., residue edge scalar features) and $f_{(i,j)}^V$ (i.e., residue edge vector features), respectively. GVP is a special neural network module which is mathematically demonstrated to be insensitive to 3D rigid motions [please refer (Jing et al., 2020) for details], and $f_{(i,j)}^{message}$ represents the message representation flowing/aggregating from residue node j to i during message passing.

C.15 Detailed description of the devised pre-training strategies

1. Backbone denoising: to make the GEE encoder better capture the geometric relationships between residue backbone atoms and potential backbone conformation change brought by different molecular generation tools, the backbone

denoising task was devised. For each protein-protein complex, we first randomly selected 15% of its residues and set a (boolean) mask to these residues, and then added noise randomly drawn from the standard normal distribution with the truncation (the maximum and minimum noise values are 2 and -2 separately) to raw backbone atom coordinates (i.e., N, $C\alpha$, C, O) of masked residues. Besides, the noise extraction for coordinate corruption is independent for each atom.

After that, all coordinates in the corrupted protein-protein complex (in the pre-training set) were normalized using its backbone atom centroid. Under the normalized coordinates, the KNN and radius contact graphs are generated, and then relevant node and edge geometric features are constructed based on the corrupted graphs. Furthermore, for the masked residues, their absolute $C\alpha$ coordinates (i.e., the $C\alpha$ coordinates before centroid-based normalization) before and after the corruption are retrieved as the denoising objective. Specifically, the final representations of masked residues produced by the GEE encoder are sent to a multi-layer perception (MLP), for predicting the relative distance between the above two sets of $C\alpha$ coordinates. The loss function measuring the denoising error is the mean square error (MSE, denoted as $MSE^{C\alpha}$).

2. Side chain denoising: to effectively capture the geometric relationships between side chain atoms and between backbone and side chain atoms on the basis of above backbone conformation change learning, for each of the aforementioned corrupted residues, we first retrieved its side chain coordinate set after the preliminary centroid normalization (i.e., $C_i^{S'}$). Then we randomly selected one of its side chain atom coordinates and added above normal distribution noise to this coordinate. Next, we calculated the centroid of side chain atoms, center of mass of side chain atoms, maximum coordinate value of side chain atoms, and unit vector of $C\alpha$-side chain geometric relationships (detailed in appendix C.13) for the corrupted residues.

   The objective for this task is to recover these side chain-related geometric features in the PPI contact graph before corruption, based on those in the contact

graph after corruption as well as other corrupted geometric features. The denoising is performed using an independent MLP predictor, and the loss function is also MSE, denoted as MSE^{Sidec}.

3. SASA prediction: other than the above four types of features that can capture side chain-related geometric properties, residue SASA values in a protein-protein complex are also able to depict such properties from a high-level perspective. Therefore, the SASA prediction task is incorporated, we first set all SASA values in the masked residues to 0, and same to above, we used the MLP and MSE loss (denoted as MSE^{SASA}) to formulate this prediction task.

4. AA type prediction: the AA type (representation) carries high-level biophysical properties of a residue, if retain it in the masked residues, the learning objectives of the above three pre-training tasks could be more easily to be achieved. In order to improve the model generalization by increasing the difficulty of the training task, for the corrupted residues, their AA type representations are also masked and to be predicted by another MLP, and the loss function used is the binary cross entropy (BCE) loss (denoted as BCE^{AA}).

C.16 Details of the interface identification based on PyMOL

In this study, we used a PyMOL script called InterfaceResidues.py to determine whether an amino acid is located at the interface of protein-protein interactions. This script identifies interface residues between two proteins or chains using the following method. First, it calculates the surface area of each amino acid of the protein complex. Then, it separates the complex into two parts, one for each chain, and calculates the surface area of each amino acid of each chain independently. Finally, it calculates the difference in the surface area of each amino acid in the complex and in each chain. This approach is widely used for identifying interface residues in protein complexes and can provide valuable insights into the molecular basis of protein-protein interactions. Therefore, we

did not need to explicitly consider the protein interface orientation when building the protein complex. The script can be found in http://www.protein.osaka-u.ac.jp/rcsfp/supracryst/suzuki/jpxtal/Katsutani/en/interface.php.

C.17 Summary of implementation tools for MpbPPI

Our basic program language is Python 3.9.13, on which Pytorch 1.12.1 (Paszke et al., 2019) with a default random seed 1234 was used to construct the framework of the GEE encoder (and corresponding pre-training strategies). For the GBT-based decoder implementation, we applied PyMOL 2.4.1 (DeLano et al., 2002) to search for the interface residues, and scikit-learn library 1.2.2 (Pedregosa et al., 2011) was adopted to establish the GBT model. For the mutant structure generation, tools or softwares to be considered include FoldX 4.0, MODELLER 10.4, and AlphaFold2. Besides, FoldX 4.0 was also used to complete the side chains for raw PDB files in both pre-training and downstream sets.

Appendix D

Supplementary for Chapter 6

D.1 The version characteristics of MARTINI22 and MARTINI3

The MARTINI force field, particularly its MARTINI22 version (De Jong et al., 2013), has been extensively employed for the coarse-grained (CG) representation of biomolecular systems (Marrink and Tieleman, 2013; Kmiecik et al., 2016). However, as the usage of MARTINI22 has increased, several shortcomings have been found, one of which is the tendency for some molecules to interact too strongly, primarily due to overly strong non-bonded interactions and the insufficient chemical space coverage brought by the relatively limited types and sizes of the CG beads (Javanainen, Martinez-Seara, and Vattulainen, 2017; Marrink et al., 2023). To alleviate these limitations, (Souza et al., 2021) developed MARTINI3 force field, which features a rebalancing of all non-bonded interaction terms in the MARTINI model and introduces new bead types and labels, providing a richer and versatile representation of molecular interactions and dynamics. In our research, we explore the integration of the traditional MARTINI22 and the latest MARTINI3 CG-scale models with graph neural networks (GNNs) for predicting protein-protein interaction (PPI) overall properties.

TABLE D.1: The statistics of nodes and edges in each set of the MANY/DC dataset.

Scale	Sub-set	Graph node number	Graph edge number	Average Node degree
CG-M2	Training	321.780	2373.738	7.377
CG-M2	Validation	321.315	2371.637	7.381
CG-M2	Test	392.377	2900.821	7.393
Scale	Sub-set	Graph node number	Graph edge number	Average Node degree
CG-M3	Training	341.764	2697.742	7.894
CG-M3	Validation	338.694	2669.420	7.882
CG-M3	Test	416.216	3298.975	7.926
Scale	Sub-set	Graph node number	Graph edge number	Average Node degree
Atom	Training	1226.182	18,897.831	15.412
Atom	Validation	1221.720	18,807.455	15.394
Atom	Test	1504.132	23,402.179	15.559
Scale	Sub-set	Graph node number	Graph edge number	Average Node degree
Residue	Training	156.996	3023.984	19.261
Residue	Validation	156.553	3016.193	19.266
Residue	Test	192.682	3772.550	19.579

D.2 The statistics of graph nodes and edges of the MANY/DC dataset

Based on the standard dimer dataset MANY/DC (Baskaran et al., 2014; Duarte et al., 2012) and conventional dataset splitting settings (Réau et al., 2023; Wang et al., 2024) as an example, we give the statistics information of nodes and edges of the protein complex graphs at different scales, in which the atom- and residue-scale complex graphs are constructed based on the default settings of atom- and residue-scale GearNet-Edge (i.e., the counterparts of MCGLPPI), respectively (Zhang et al., 2022; Zhang et al., 2024). Additionally, to differentiate the results of MCGLPPI supported by MARTINI22 and MARTINI3, we denote their corresponding scale names as "CG-M2" and "CG-M3", respectively (table D.1).

We can first observe that, compared with the atom-scale counterpart based on the full-atom characterizations, the CG-scale protein complex graph produced by MCGLPPI has overall smaller graph size: the number of nodes and edges of the atom-scale graph, are approximately three times and seven times greater, respectively, than those at the CG-scale graph.

Meanwhile, for the residue-based counterpart, its modern design of graphs for protein property predictions like residue-scale GearNet-Edge is more relying on the construction of edges, which are usually fully built based on multiple pre-defined

geometric distance and sequential thresholds, aiming to capture more comprehensive spatial relationships between residue nodes (Zhang et al., 2022; Zhang et al., 2024; Zeng et al., 2023). While the number of edges will significantly influence the neighboring message aggregation (Kipf and Welling, 2016) speed of corresponding graph neural networks. In contrast, our CG graph adopts more chemical-plausible edges according to the MARTINI force field. For these MARTINI-based edges, they are constructed based on the specific interaction definitions between designated bead node pairs, thereby reducing the reliance on indiscriminately wiring every node pair within multiple pre-defined thresholds. Under current settings, although the average graph node number in CG-scale is relatively higher, the average graph edge number and node degree are both observed a significant decrease, which is ultimately beneficial to overall more sparse and accurate edge distribution and thus contributes to effectively decreasing the processing speeds.

Besides, due to the moderate extension of bead types and numbers of MARTINI3, its corresponding graph size is slightly larger than that from MARTINI22.

D.3 Further extension to protein complex $\Delta\Delta G$ predictions based on simple modifications

we demonstrate the potential of our method MCGLPPI on the extension into directly predicting $\Delta\Delta G$ with the simple modifications. Specifically, we use a multiple-point mutation dataset AB-bind (Sirin et al., 2016), which contains 1101 sample points related to the binding affinity change (i.e., $\Delta\Delta G$) caused by multiple-point AA mutations on the complex formed from antibody or antibody-like binding. Following the basic experimental settings of the existing pre-trained GNN-based approach MpbPPI (Yue et al., 2023c), which has already tested on this dataset, FoldX (Schymkowitz et al., 2005) is chosen to complete all side chains and generate the mutation complex structure based on the corresponding raw PDB file of wild-type (WT) structure and mutational site information (and then all structures will be transformed into CG-scale protein graphs based on MARTINI). Based on this, two groups of different validation experiments are conducted as follows.

For the first group of experiments, we first consider a WT protein-protein complex type-based fivefold cross-validation (CV) evaluation setting. This setting reduces the sample similarity between training and test sets by ensuring no intersection of original WT protein-protein complex types [for corresponding WT-mutant (MT) pairs] between any of the two folds and striving to make the both sample total number and WT protein-protein complex type number assigned for each fold to be as close as possible [detailed in original MpbPPI paper (Yue et al., 2023c)]. For the related model modification details, $\Delta\Delta G$ represents the binding affinity (i.e., binding free energy, ΔG) change from WT to MT status (i.e., $\Delta G^{WT} - \Delta G^{MT}$). Inspired by this (Yue et al., 2023c; Liu et al., 2021), keeping other model configurations unchanged, the $\Delta\Delta G$ calculation is training optimized and then predicted based on a three-layer multi-layer perception (MLP) which receives the difference between graph embeddings of the CG complex graph before mutation and after mutation (i.e., $\mathbf{h}^{WT} - \mathbf{h}^{MT}$) provided by the protein encoder initiailized from the pre-trained model parameter checkpoint.

We evaluate our method (MCGLPPI-M2), and record the prediction results based on the R_P, RMSE, and MAE. We also report the corresponding results of MpbPPI and two representative energy-based specialized methods FoldX and Rosetta macromolecular modeling suite (Flex ddG) (Barlow et al., 2018) from (Yue et al., 2023c) (table D.2). From these experimental results, we find that under current settings, our method achieves competitive performance compared with the advanced $\Delta\Delta G$ prediction methods. It is worth noting that our method only adopts a straightforward adaptation without any specifically designed modules, while the compared methods are carefully proposed for the $\Delta\Delta G$ predictions.

At the same time, based on the same model modifications, we further test the stability of our method through a data split with the explicit TM-score cutoff. Specifically, we utilize the TM-align alignment tool to calculate a similarity matrix based on the structural difference between pairwise PDB samples within AB-bind, and extract sample points whose TM-score is lower than 0.5 with other PDB samples as the test set (183 sample points are found in total), and the remaining ones are treated as the training set. In this setting, MCGLPPI still can give a test R_P over 0.4 (table D.2).

[1]The bold data signifies the best experimental result under the current comparison group.

TABLE D.2: the $\Delta\Delta G$ prediction results based on the AB-bind dataset.

5-fold WT type-based CV	R_P	RMSE	MAE
MCGLPPI	**0.451**	**1.880**	1.363
MpbPPI (Yue et al., 2023c)	0.442	1.899	**1.357**
FoldX (Schymkowitz et al., 2005)	0.273	3.429	2.278
Flex ddG (Barlow et al., 2018)	0.059	4.494	2.875
TM-score splitting	R_P	RMSE	MAE
MCGLPPI	0.415	1.327	1.081
AB-bind 962-subset inference	R_P	RMSE	MAE
MCGLPPI	**0.280**	**2.420**	**1.563**
GearNet-Atom (Zhang et al., 2024)	0.150	2.464	1.606
GearNet-Res (Zhang et al., 2022)	0.074	2.476	1.609
AB-bind complete inference	R_P	RMSE	MAE
MCGLPPI	0.264	2.386	1.511

1

On top of this, another group of experiments are conducted about investigating the feasibility of directly using the trained ΔG models for $\Delta\Delta G$ predictions according to the existing data. Specifically, we train MCGLPPI-M2, GearNet-Atom, and GearNet-Res based on the 915-sample-subset of the independent PDBbind-strict-dimer ΔG dataset (see section 6.2.6), and predict $\Delta\Delta G$ for all samples in the 1101-sample AB-bind dataset based on the prediction of ΔG^{WT} minus the prediction of ΔG^{MT} without extra training (table D.2, please note that for GearNet-Atom and GearNet-Res, 962 out of all test samples can be identified and inferred). We observe that, although our model is trained with an independent ΔG dataset that does not include specialized WT-MT antigen-antibody pairs with explicit $\Delta\Delta G$ labels to guide the model to directly perceive the subtle binding intensity differences between similarity WT and MT structures, MCGLPPI is still able to give a reasonable predictive accuracy competitive to the specifically designed energy-based prediction tool FoldX. Furthermore, it also outperforms its atom- and residue-scale counterparts. In summary, the aforementioned experiments further demonstrate the advantages of our method and prove the flexibility and versatility of the CG-scale framework in handling various PPI property prediction tasks.

D.4 Detailed curation process of the PDBbind-strict-dimer dataset

We retrieve 2852 protein–protein complexes with known binding affinity data in total from the PDBbind database (version 2020). Initially, we refine this dataset to include only the simplest types of complexes, which are composed of two protein components. We then select those samples that form a single PPI binding interface, as determined by their three-dimensional (3D) structural configurations. In cases where a two-component complex has multiple PPI binding interfaces, but these interfaces are structurally similar, we retain the sample and extract the structural information for the two proteins forming one representative binding interface. These filtering procedures result in a strict dataset of 1270 dimeric complexes (termed as the PDBbind-strict-dimer dataset).

D.5 The default hyper-parameters of involved approaches in comparison

We compare our CG-scale MCGLPPI framework with the counterpart GearNet-Edge which can function at the atom- or residue- scales (Zhang et al., 2022; Zhang et al., 2024). We also incorporate an atom-scale advanced approach GVP-GNN (Jing et al., 2021), which is specifically designed to learn 3D macromolecular structures (esp., protein-protein complexes), into the complete comparison experiments. We give the basic default hyper-parameters of these involved approaches from original papers (Zhang et al., 2022; Zhang et al., 2024; Jing et al., 2021) as follows.

1. MCGLPPI (CG-scale, applicable to both MARTINI22- and MARTINI3-based versions): The detailed graph construction settings are provided in Methods of the manuscript. The default CG-scale graph convolutional layer number and hidden feature dimension are 6 and 256, respectively.

2. GearNet-Edge (atom-scale) (Zhang et al., 2022; Zhang et al., 2024): The construction of protein graph structures for GearNet-Edge at the atom-scale is based on the radius edge: each atom node i will connect all other atom nodes

within a 4.5Å-radius 3-dimensional (3D) sphere centered at i. The default graph convolutional layer number and hidden feature dimension are 6 and 128, respectively.

3. GearNet-Edge (residue-scale) (Zhang et al., 2022; Zhang et al., 2024): The construction of protein graph structures for GearNet-Edge at the residue-scale is built based on three different types of distance-based or sequence-based edges.

    - Radius edge: Analogous to those used in GearNet-Edge (atom-scale), the only difference is that the radius cutoff is set to 10 Å when wiring the neighboring residue nodes.
    - KNN edge: Each residue node i connects other 10 nodes nearest to it, based on the Euclidean distance between their node position coordinates (and an extra 5 Å cutoff is imposed for limiting the minimum inter-residue distance).
    - Sequential edge: Two residue nodes will be wired if these two nodes have the relative positional difference N in the amino acid (AA) sequence of current protein. The pre-defined N includes 1, 2, and 3.

    Besides, The default graph convolutional layer number and hidden feature dimension are 6 and 512, respectively.

4. GVP-GNN (atom-scale) (Jing et al., 2021): Based on the same radius edge setting as GearNet-Edge at the atom-scale (4.5Å cutoff), the graph structure is constructed. GVP-GNN explicitly distinguishes scalars and vectors within the node and edge features. For the default hidden dimensions for node features, 100 (for scalars) and 16 (for vectors) are adopted. For the hidden dimensions of edge features, 32 (for scalars) and 1 (for vectors) are chosen. Besides, its default graph convolutional layer number is set to 5.

D.6 Investigation about the influence of hidden feature dimensions on overall performance of MCGLPPI

Because the default hidden feature dimensions of MCGLPPI and its atom- and residue-scale counterparts (i.e., GearNet-Edge) are 256, 128, and 512, respectively (the default graph convolutional layer number is the same: 6). In order to investigate the robustness of our MCGLPPI to the default dimensions of its atom- and residue-scale counterparts, we conduct an extra experiment as follows.

Specifically, based on the experimental results on 915-subset of the PDBbind-strict-dimer dataset (see table 6.1), we can find the top predictive performance of atom- and residue-scale GearNet-Edge is both achieved at batch size equaling to 16. Therefore, we separately adjust the hidden feature dimension of MCGLPPI (MCGLPPI-M2 is used here) into 128 and 512 at the batch size 16 to conduct a fair comparison, and the results under the corresponding settings are recorded.

For the combination of hidden feature dimension 128 plus batch size 16, the results are: 0.587 (R_P), 2.088 (RMSE), 1.615 (MAE), 1625 (GPU (MB)), and 9818 (Total Time (s)). The corresponding results under the combination of hidden feature dimension 512 plus batch size 16 are: 0.579 (R_P), 2.079 (RMSE), 1.614 (MAE), 6658 (GPU (MB)), and 24,893 (Total Time (s)). We can observe that, after keeping the same hidden feature dimensions as the atom- and residue- counterparts, MCGLPPI still achieves the competitive predictive performance while preserving relatively lower computational cost (refer table 6.1, and the optimal results of MCGLPPI are achieved at a combination of hidden feature dimension 256 with batch size 64). This can validate the robustness of MCGLPPI to the hidden feature dimension, and further demonstrate the effectiveness of our CG-scale protein geometric representation learning framework.

D.7 The experimental results on more challenging experimental settings

Based on the PDBbind dataset, we conduct an additional more challenging experiment based on further structural homology reduction. Specifically, for the PDBbind

TABLE D.3: The experimental results based on structural homology reduction settings.

W/ pre-training	Scale	R_P	RMSE	MAE
MCGLPPI-M2	CG	**0.449**	**2.084**	**1.666**
MCGLPPI-M3	CG	0.442	2.503	1.966
GearNet-Atom (Zhang et al., 2024)	Atom	0.384	2.287	1.817
GearNet-Res (Zhang et al., 2022)	Residue	0.369	2.370	1.857
W/O pre-training	Scale	R_P	RMSE	MAE
MCGLPPI-M2	CG	0.406	**2.111**	**1.723**
MCGLPPI-M3	CG	**0.426**	2.237	1.745
GearNet-Atom (Zhang et al., 2024)	Atom	0.346	2.276	1.771
GearNet-Res (Zhang et al., 2022)	Residue	0.365	2.541	2.037
GVP-GNN (Jing et al., 2021)	Atom	0.221	3.835	3.231

[2]

915-subset for which all involved methods can completely identified, we perform the pairwise TM-align structural alignments (Zhang and Skolnick, 2005) against all samples in this 915-subset, resulting in a 915×915 TM-score matrix with each element representing the structural similarity between the corresponding samples. Furthermore, for each sample, we choose the maximum TM-score in the corresponding row of the matrix, i.e., its maximum structural similarity against all other samples in the dataset. On top of this, we extract samples with maximum TM-score lower than 0.45 as the test set, representing the samples with the lowest homology structure similarities (with other all samples) in the dataset (124 samples in total). The samples with the TM-score ranging from 0.45-0.55 are used as the validation set (85 samples in total), and the rest of samples are treated as the training set (706 samples).

Based on the aforementioned experimental settings, we select the best basic model settings under the corresponding tenfold cross-validation (CV) settings for each involved method (including those with and without pre-training), to conduct the comparison (table D.3). The experimental results indicate that, MCGLPPI-M2 and MARTINI-M3 brings significant performance improvements on Pearson's correlation coefficient (R_P) under this challenging test scenario, further demonstrating the generalization ability of the proposed framework.

[2]The bold data signifies the best experimental result under the current comparison group.

D.8 Model stability test using more accessible AlphaFold-generated structures

To examine the binding affinity prediction stability of models in the scenario in which the expensive real structures are not available, based on the PDBbind dataset, we conduct an extra experiment as follows.

We first generate the protein complex structures for samples in the PDBbind-strict-dimer dataset using the publicly available ColabFold (Mirdita et al., 2022), a simplified version based on AlphaFold (Jumper et al., 2021) and AlphaFold-multimer (Evans et al., 2021). The FASTA sequence file containing the complete sequences of both proteins in each dimer is used as input for the ColabFold. Following the ColabFold pipeline, we perform a MSA search with MMseqs2 on the UniRef30 database and generate five models for each sample. These models are subsequently relaxed with the Amber program and ranked. We ultimately select the top-ranked structure as the AlphaFold generated structure for each sample in the PDBbind-strict-dimer dataset. Then, the generated complex structures are transformed into the CG-graphs, and their atom-scale and residue-scale counterparts.

After that, 911 of 1270 samples are successfully identified by the atom- and residue-scale GearNet-Edge (which are also covered by the PDBbind 915-subset). Based on this 911-subset, we perform the comparison based on two different sample splitting settings, i.e., the standard tenfold CV and the TM-score-based splitting (702-85-124 samples for training, validation, test sets, detailed description is provided in appendix D.7), and record the experimental results based on the best model configurations under the real structure testing (table D.4).

We observe that, under current experimental settings, the models trained with more accessible AlphaFold-generated structures achieve close predictive accuracy compared with those derived from real structures, indicating that when feeding the AlphaFold-generated structures, the models are also capable of adapting to their structural patterns to give reasonable predictions, which is valuable when the expensive real structures are difficult to acquire.

[3]The bold data signifies the best experimental result under the current comparison group.

TABLE D.4: The experimental results based on AlphaFold-generated structures.

10-fold CV	Scale	R_P	RMSE	MAE
MCGLPPI-M2	CG	0.589	2.088	1.572
MCGLPPI-M3	CG	**0.599**	**2.084**	**1.566**
GearNet-Atom (Zhang et al., 2024)	Atom	0.526	2.331	1.731
GearNet-Res (Zhang et al., 2022)	Residue	0.567	2.156	1.623
TM-score splitting	Scale	R_P	RMSE	MAE
MCGLPPI-M2	CG	0.404	**2.130**	**1.736**
MCGLPPI-M3	CG	**0.409**	2.970	2.378
GearNet-Atom (Zhang et al., 2024)	Atom	0.292	2.249	1.803
GearNet-Res (Zhang et al., 2022)	Residue	0.363	2.261	1.760

[3]

D.9 The other experimental results for the evaluation of the DDI-based pre-training technique

Following the same experimental settings described in section 6.2.9, we here provide the predictive performance of the pre-trained MCGLPPI-M2, MCGLPPI-M3, and its atom- and residue-scale counterparts, which are evaluated based on the other metrics (except for R_P and AUPR) on the downstream datasets.

From the experimental results (table D.5), we find that under the current pre-training settings and evaluation metrics, the MCGLPPI-M2 and MCGLPPI-M3 still outperform their atom- and residue-scale counterparts, which is in line with the results shown in figure 6.3 (a).

In addition, under the full set of 3DID pre-training set, for MCGLPPI(-M2), the R_P on the PDBbind dataset, R_P on the ATLAS dataset, and AUPR on the MANY/DC dataset are 0.610, 0.836, and 0.865, respectively.

D.10 More specific performance difference analysis across different MARTINI versions

When training from scratch, in the task of predicting PPI affinities on the PDBbind and ATLAS datasets, MCGLPPI-M3 outperforms MCGLPPI-M2. However, in the task of classifying complex interface structures on the MANY/DC dataset, the performance of MCGLPPI-M3 and MCGLPPI-M2 is comparable.

[4]The bold data indicates the best experimental result under current dataset and evaluation metric.

TABLE D.5: The other experimental results of the pre-training at different scales.

PDBbind				
Pre-training set	Model name	Scale	RMSE	MAE
Complete 3did	MCGLPPI-M2	CG	**2.037**	1.572
The 33144-set	MCGLPPI-M2	CG	2.053	1.588
The 33144-set	MCGLPPI-M3	CG	2.048	**1.569**
The 33144-set	GearNet-Edge	Atom	2.163	1.646
The 33144-set	GearNet-Edge	Residue	2.165	1.632
ATLAS				
Pre-training set	Model name	Scale	RMSE	MAE
Complete 3did	MCGLPPI-M2	CG	**0.998**	0.765
The 33144-set	MCGLPPI-M2	CG	1.002	**0.745**
The 33144-set	MCGLPPI-M3	CG	1.024	0.778
The 33144-set	GearNet-Edge	Atom	1.052	0.771
The 33144-set	GearNet-Edge	Residue	1.056	0.808
MANY/DC				
Pre-training set	Model name	Scale	AUROC	
Complete 3did	MCGLPPI-M2	CG	0.874	
The 33144-set	MCGLPPI-M2	CG	**0.877**	
The 33144-set	MCGLPPI-M3	CG	0.860	
The 33144-set	GearNet-Edge	Atom	0.838	
The 33144-set	GearNet-Edge	Residue	0.855	

This variation in performance could be attributed to the fact that binding energy prediction is closely related to quantifiable physical parameters such as bead distance and interface charge (Kortemme, Morozov, and Baker, 2003). MARTINI3 provides more appropriate bead segmentation and types than MARTINI22 (Souza et al., 2021), leading to more precise descriptions of physical contacts and energy characterization. Consequently, the performance of MCGLPPI based on MARTINI3 tends to be superior in PPI affinity predictions. However, classifying PPI interfaces—involving biological or crystallographic classification—is more complex than free energy calculation, containing factors from the atomic level to the entire biological system level, which may explain the negligible performance differences between MCGLPPI-M2 and MCGLPPI-M3. Although MCGLPPI based on MARTINI3 outperforms its MARTINI2-based counterpart in terms of accuracy (in predicting binding affinities), the latter enjoys a slightly faster processing speed due to its reduced number of beads and bonds. This trade-off highlights the balance between computational efficiency and predictive accuracy within different versions of the MARTINI-based MCGLPPI models.

Furthermore, we also find that the use of DDIs-based pre-training not only improves the overall accuracy but also helps to minimize the performance differences between MCGLPPI-M2 and MCGLPPI-M3.

D.11 The potential extension of using corresponding force field parameters in other scales

In addition the CG-scale force field, all-atom force fields such as CHARMM (Huang et al., 2017) and AMBER (Salomon-Ferrer, Case, and Walker, 2013) provide similar bonded and nonbonded parameters, which can be technically integrated with corresponding atom-scale GNNs to predict PPI properties. However, the number of edges in an atom-scale graph constructed from all-atom force field parameters exceeds those in a MARTINI-based CG-scale graph by more than three times. For instance, in the classical Barnase-Barstar protein complex, the AMBER99SB-ILDN force field (Smith et al., 2015) includes 1616 heavy atom bond parameters for graph edges, far exceeding the 495 and 521 bond parameters provided by the MARTINI22

and MARTINI3 force fields, respectively. Moreover, all-atom force fields offer a wider variety of atom types, angles, and dihedral parameters, indicating that GNN models based on all-atom fields are more complex and computationally demanding relative to their CG-scale counterparts. Furthermore, there are some simplified models where each protein residue is represented by a single bead, typically positioned at the $C\alpha$ location (Bahar and Jernigan, 1997; Clementi, Nymeyer, and Onuchic, 2000). While these models are suited for simulating specific systems, they lose the granularity of the side chains and local structural nuances, facing difficulties in providing accurate and unbiased structural representations of more complex protein features when using one-bead-per-residue force field parameters. This limitation highlights the challenges involved in integrating residue-based force fields with corresponding GNNs. In total, the proposed MCGLPPI, which is based on the MARTINI force field, strikes a balance between detailed structural representations of proteins and acceptable computational efficiency.

D.12 The definition of generation of a one-hot representation

"One-hot" representation refers to transforming the categorical scalar value from current feature (e.g., bead type of MARTINI22 or MARTINI3) into a unique binary representation, with the size equaling to the number of total categories appearing in current feature (e.g., 17 for bead type of MARTINI22 or 23 for bead type of MARTINI3). In this representation, the position corresponding to the current category is marked as 1, while all other positions are marked as 0.

D.13 Detailed description of allocation of sparse angular features to specific bead nodes

For each angular parameter generated by the MARTINI22 or MARTINI3 engines, its sine-cosine encoded feature will be allocated to specific bead node based on the following rules:

- The backbone angles (θ_{BBB}): For each θ_{BBB}, the corresponding encoded feature will be assigned to the second bead node of current BBB combination.
- The backbone-side chain angles (θ_{BBS}): For each θ_{BBS}, the corresponding encoded feature will be assigned to the third bead node of current BBS combination.
- The side chain angles (θ_{BSS}): For each θ_{BSS}, the corresponding encoded feature will be assigned to the third bead node of current BSS combination.
- The backbone dihedrals (Ψ_{BBBB}): For each Ψ_{BBBB}, the corresponding encoded feature will be assigned to the second bead node of current $BBBB$ combination.

D.14 The equations of the GearNet-Edge protein encoder

Concisely, GearNet-Edge introduces a line graph-augmented edge message passing strategy (Harary and Norman, 1960), which models the inter-edge relative positional relationship, to inject the additional structural information into the node representations for more effective protein geometric characteristics learning. The relevant equations are as follows:

$$\mathbf{h}_i^0 = \mathbf{f}_i \tag{D.1}$$

$$\mathbf{h}_i^l = \mathbf{h}_i^{l-1} + \mathbf{u}_i^l \tag{D.2}$$

$$\mathbf{u}_i^l = \sigma(BN(\sum_{r\in\mathcal{R}} W_r \sum_{j\in N_r(i)} (\mathbf{h}_j^{l-1} + MLP(\mathbf{m}_{(i,j,r)}^l)))) \tag{D.3}$$

$$\mathbf{m}_{(j,i,r_1)}^0 = \mathbf{f}_{(j,i,r_1)} \tag{D.4}$$

$$\mathbf{m}_{(j,i,r_1)}^l = \sigma(BN(\sum_{r\in\mathcal{R}'} W_r' \sum_{(k,w,r_2)\in N_r'((j,i,r_1))} \mathbf{m}_{(k,w,r_2)}^{l-1})) \tag{D.5}$$

in which $\mathbf{f}_i$ and $\mathbf{f}_{(j,i,r)}$ are initial node and edge features, $\mathbf{h}_i^l$ and $\mathbf{u}_i^l$ are the hidden representation of node i at the l-th (graph convolutional) layer and aggregated neighboring information of node i at the l-th layer, respectively. For the calculation of $\mathbf{u}_i^l$ (equation (D.3)), σ, BN, W_r, $N_r(i)$, MLP, and $\mathbf{m}_{(i,j,r)}^l$ represent the Sigmoid

activation function, Batch Normalization layer (Santurkar et al., 2018), edge-type-specific linear transformation matrix, 1-hop neighbors of node i with edge type r, multi-layer perception (MLP), and augmented edge feature based on the line graph at l-th layer, respectively. Furthermore, for the calculation of the augmented edge feature (equation (D.5)), it follows the similar information aggregation mechanism as equation (D.3), which will aggregate the neighboring edges' information into the feature of the central edge based on a pre-defined line graph. For the description of the line graph construction among existing edges, please refer (Zhang et al., 2022) for detailed information.

D.15 The details about the CG-scale pre-training technique

For every protein domain-domain (DDI) complex curated from the 3DID dataset, we first transform it into a CG-scale complex graph following the same procedure as that for downstream complex transformations. These CG-scale DDI complex graphs will then be leveraged by a CG-scale diffusion denoising pre-training technique developed based on the atom-scale work (Zhang et al., 2024), aiming to make the CG graph encoder aware the general DDI knowledge from these samples (these operations are applicable for both MARTINI22- and MARTINI3-based versions).

Specifically, by jointly modelling of coarse-grained protein complex conformation and underlying sequence information, the more comprehensive learning of CG DDI complex structures could be achieved. Based on this, the diffusion mechanism (Ho, Jain, and Abbeel, 2020) is used to naturally add noise with varying magnitudes into the 3D coordinates and sequences of bead nodes within CG-scale DDI complex graphs to corrupt these graphs, and then to force the CG protein encoder to recover the original conformation and sequence for the complete joint modelling. The whole process is formulated as follows, equations (D.6) and (D.7) represent the noising adding and denoising processes respectively.

$$q(D^{1:T} \mid D^0) = \prod_{t=1}^{T} q(D^t \mid D^{t-1}) \tag{D.6}$$

$$p_\theta(D^{0:T-1} \mid D^T) = \prod_{t=1}^{T} p_\theta(D^{t-1} \mid D^t) \tag{D.7}$$

in which the both processes can be decomposed into multiple time steps of noising adding or denoising following the rule of Markov chains (i.e., the total time steps are T). For each step t, $q(D^t \mid D^{t-1})$ and $p_\theta(D^{t-1} \mid D^t)$ define the current noising adding and denoising based on the parameterized CG graph encoder, respectively (in which D^t represents the status of the noised CG DDI complex (graph) at step t). Additionally, $q(D^t \mid D^{t-1})$ and $p_\theta(D^{t-1} \mid D^t)$ can be further decomposed into the CG conformation and bead sequence noising adding and denoising processes, to realize the aforementioned joint modelling [equations (D.8) and (D.9)].

$$q(D^t \mid D^{t-1}) = q(S^t \mid S^{t-1}) \cdot q(C^t \mid C^{t-1}) \tag{D.8}$$

$$p_\theta(D^{t-1} \mid D^t) = p_\theta(S^{t-1} \mid D^t) \cdot p_\theta(C^{t-1} \mid D^t) \tag{D.9}$$

in which C^t and S^t denote the status the of CG DDI complex conformation and bead sequence at step t, respectively.

For $q(S^t \mid S^{t-1})$ that is performed prior to the $q(C^t \mid C^{t-1})$ in each step of noising adding, all side chain bead nodes S (and their corresponding edges and features), which belong to the specified amino acids (AAs) to be masked, will be cropped out from the current CG DDI complex graph (to get a new cropped graph). This is for masking the sequence information that can indicate the AA types based on the pre-defined AA masking ratio value ρ^t. After that, the independent coordinate noise drawn based on the Gaussian distribution (controlled by the pre-defined distribution variance value β^t) will be injected into the 3D coordinate of every bead node within current graph after the above cropping, to corrupt the overall CG-scale conformation for finishing current noising adding step.

With regards to the corresponding denoising process $p_\theta(D^{t-1} \mid D^t)$ for CG-scale conformation and sequence recovery, the cropped graph produced based on

$q(D^t \mid D^{t-1})$ is sent to the CG graph encoder to generate the node-level geometric representations for current corrupted CG DDI complex. These representations will be further sent to the structural denoising network and sequential denoising network, for predicting 1) the CG-scale conformation change (brought by CG conformation noise adding) measured by the distance between pairwise bead nodes of edges within the cropped graph, and 2) the AA types which are masked in CG sequence noising adding, respectively. The mean square error (MSE, denoted as L^C) and the cross-entropy (CE, denoted as L^S) are chosen as the loss function to measure the respective differences between the recovered values and corresponding ground truths. The total loss to guide the pre-training model optimization can thus be formulated as follows, in which α is the loss weight to balance the both denoising process.

$$L^{pre-training} = \alpha L^S + (1-\alpha)L^C \tag{D.10}$$

For the aforementioned AA masking ratio ρ^t, distribution variance β^t, structural denoising network, and sequential denoising network, we follow the settings in (Zhang et al., 2024), we suggest referring the original paper for the detailed description. On top of this, the pre-training is performed based on the CG graph encoder on the curated CG-scale 3DID pre-training set under 200 epochs with $\alpha = 0.5$, followed by additional 50 epochs $\alpha = 0.8$. In addition, another difference compared with the atom-scale work in (Zhang et al., 2024) is that, we do not incorporate the similar conformer generation mechanism, which aims to bring more conformation varieties of original proteins, into the CG-based diffusion training process, for which we find that it does not help to the further performance improvement for our CG-scale geometric learning framework.

D.16 Summary of implementation tools for MCGLPPI

Our basic program language is Python 3.9.18, on which Pytorch 1.12.1 (Paszke et al., 2019) and Torchdrug (Zhu et al., 2022) 0.2.1 with a default random

seed 0 are used to construct the overall framework of the MCGLPPI (including MCGLPPI-M2 and MCGLPPI-M3). For the CG structure and CG-scale force field parameter generation, the Python-based script martinize.py (https://cgmartini.nl/docs/downloads/tools/proteins-and-bilayers.html, version 2.4) (De Jong et al., 2013) (for MARTINI22) as well as the Martinize2 and Vermouth programs (Kroon et al., 2022) (for MARTINI3) are used for the transformation process. Besides, before the parameter generation, we adopt the pdbfixer tool (https://github.com/openmm/pdbfixer) to complete missing side-chains and convert non-natural amino acids to their natural counterparts.

Bibliography

Abramson, Josh et al. (2024). "Accurate structure prediction of biomolecular interactions with AlphaFold 3". In: *Nature*, pp. 1–3.

Agalliu, Ilir et al. (2015). "Higher frequency of certain cancers in LRRK2 G2019S mutation carriers with Parkinson disease: a pooled analysis". In: *JAMA neurology* 72.1, pp. 58–65.

Alberts, Bruce (2002). "Molecular biology of the cell 4th edition". In: *(No Title)*.

Alberts, Bruce et al. (2002). "Analyzing protein structure and function". In: *Molecular Biology of the Cell. 4th edition*. Garland Science.

Alborzi, Seyed Ziaeddin et al. (2021). "PPIDomainMiner: Inferring domain-domain interactions from multiple sources of protein-protein interactions". In: *PLOS Computational Biology* 17.8, e1008844.

Arts, Marloes et al. (2023). "Two for one: Diffusion models and force fields for coarse-grained molecular dynamics". In: *Journal of Chemical Theory and Computation* 19.18, pp. 6151–6159.

Ashburner, Michael et al. (2000). "Gene ontology: tool for the unification of biology". In: *Nature genetics* 25.1, pp. 25–29.

Atz, Kenneth, Francesca Grisoni, and Gisbert Schneider (2021). "Geometric deep learning on molecular representations". In: *Nature Machine Intelligence* 3.12, pp. 1023–1032.

Bahar, Ivet and Robert L Jernigan (1997). "Inter-residue potentials in globular proteins and the dominance of highly specific hydrophilic interactions at close separation". In: *Journal of molecular biology* 266.1, pp. 195–214.

Balazevic, Ivana, Carl Allen, and Timothy Hospedales (2019). "Multi-relational poincaré graph embeddings". In: *Advances in Neural Information Processing Systems* 32.

Barderas, Rodrigo et al. (2008). "Affinity maturation of antibodies assisted by in silico modeling". In: *Proceedings of the National Academy of Sciences* 105.26, pp. 9029–9034.

Barlow, Kyle A et al. (2018). "Flex ddG: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation". In: *The Journal of Physical Chemistry B* 122.21, pp. 5389–5399.

Barouch, Dan H et al. (2013). "Therapeutic efficacy of potent neutralizing HIV-1-specific monoclonal antibodies in SHIV-infected rhesus monkeys". In: *Nature* 503.7475, pp. 224–228.

Baskaran, Kumaran et al. (2014). "A PDB-wide, evolution-based assessment of protein-protein interfaces". In: *BMC structural biology* 14, pp. 1–11.

Bécigneul, Gary and Octavian-Eugen Ganea (2018). "Riemannian adaptive optimization methods". In: *arXiv preprint arXiv:1810.00760*.

Belkin, Mikhail and Partha Niyogi (2001). "Laplacian eigenmaps and spectral techniques for embedding and clustering". In: *Advances in neural information processing systems* 14.

Ben-Kasus, Tsipi et al. (2007). "Cancer therapeutic antibodies come of age: targeting minimal residual disease". In: *Molecular oncology* 1.1, pp. 42–54.

Besharatifard, Milad and Fatemeh Vafaee (2024). "A review on graph neural networks for predicting synergistic drug combinations". In: *Artificial Intelligence Review* 57.3, p. 49.

Bliss, Chester I (1939). "The toxicity of poisons applied jointly 1". In: *Annals of applied biology* 26.3, pp. 585–615.

Bordes, Antoine et al. (2013). "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems* 26.

Borrman, Tyler et al. (2017). "ATLAS: a database linking binding affinities with structures for wild-type and mutant TCR-pMHC complexes". In: *Proteins: Structure, Function, and Bioinformatics* 85.5, pp. 908–916.

Braun, Pascal and Anne-Claude Gingras (2012). "History of protein–protein interactions: From egg-white to complex networks". In: *Proteomics* 12.10, pp. 1478–1498.

Bryant, Patrick, Gabriele Pozzati, and Arne Elofsson (2022). "Improved prediction of protein-protein interactions using AlphaFold2". In: *Nature communications* 13.1, p. 1265.

Burley, Stephen K et al. (2017). "Protein Data Bank (PDB): the single global macromolecular structure archive". In: *Protein crystallography: methods and protocols*, pp. 627–641.

Cai, Huiyu et al. (2024). "Pretrainable geometric graph neural network for antibody affinity maturation". In: *Nature Communications* 15.1, p. 7785.

Chami, Ines et al. (2019). "Hyperbolic graph convolutional neural networks". In: *Advances in neural information processing systems* 32.

Chang, Chia-En, Wei Chen, and Michael K Gilson (2005). "Evaluating the accuracy of the quasiharmonic approximation". In: *Journal of Chemical Theory and Computation* 1.5, pp. 1017–1028.

Chen, Bolin et al. (2014). "Identifying protein complexes and functional modules—from static PPI networks to dynamic PPI networks". In: *Briefings in bioinformatics* 15.2, pp. 177–194.

Chen, Fu et al. (2016a). "Assessing the performance of the MM/PBSA and MM/GBSA methods. 6. Capability to predict protein–protein binding free energies and re-rank binding poses generated by protein–protein docking". In: *Physical Chemistry Chemical Physics* 18.32, pp. 22129–22139.

Chen, Huiling and Huan-Xiang Zhou (2005). "Prediction of interface residues in protein–protein complexes by a consensus neural network method: test against NMR data". In: *Proteins: Structure, Function, and Bioinformatics* 61.1, pp. 21–35.

Chen, Huiyuan and Jing Li (2019). "Modeling relational drug-target-disease interactions via tensor factorization with multiple web sources". In: *The World Wide Web Conference*, pp. 218–227.

— (2020). "Learning data-driven drug-target-disease interaction via neural tensor network". In: *International joint conference on artificial intelligence (IJCAI)*.

Chen, Jun, Azza Althagafi, and Robert Hoehndorf (2021). "Predicting candidate genes from phenotypes, functions and anatomical site of expression". In: *Bioinformatics* 37.6, pp. 853–860.

Chen, Weize et al. (2021). "Fully hyperbolic neural networks". In: *arXiv preprint arXiv:2105.14686*.

Chen, Xing et al. (2016b). "Drug–target interaction prediction: databases, web servers and computational models". In: *Briefings in bioinformatics* 17.4, pp. 696–712.

Cheng, Hua et al. (2014). "ECOD: an evolutionary classification of protein domains". In: *PLoS computational biology* 10.12, e1003926.

Cho, Kyunghyun (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Chu, Zhaoyang et al. (2022). "Hierarchical graph representation learning for the prediction of drug-target binding affinity". In: *Information Sciences* 613, pp. 507–523.

Clementi, Cecilia, Hugh Nymeyer, and José Nelson Onuchic (2000). "Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? An investigation for small globular proteins". In: *Journal of molecular biology* 298.5, pp. 937–953.

Corominas-Murtra, Bernat et al. (2013). "On the origins of hierarchy in complex networks". In: *Proceedings of the National Academy of Sciences* 110.33, pp. 13316–13321.

Csermely, Peter et al. (2013). "Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review". In: *Pharmacology & therapeutics* 138.3, pp. 333–408.

Das, Rhiju and David Baker (2008). "Macromolecular modeling with rosetta". In: *Annu. Rev. Biochem.* 77.1, pp. 363–382.

Davis, Allan Peter et al. (2019). "The comparative toxicogenomics database: update 2019". In: *Nucleic acids research* 47.D1, pp. D948–D954.

De Jong, Djurre H et al. (2013). "Improved parameters for the martini coarse-grained protein force field". In: *Journal of chemical theory and computation* 9.1, pp. 687–697.

Dehouck, Yves et al. (2013). "BeAtMuSiC: prediction of changes in protein–protein binding affinity on mutations". In: *Nucleic acids research* 41.W1, W333–W339.

DeLano, Warren L et al. (2002). "Pymol: An open-source molecular graphics tool". In: *CCP4 Newsl. Protein Crystallogr* 40.1, pp. 82–92.

Dettmers, Tim et al. (2018). "Convolutional 2d knowledge graph embeddings". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1.

Devlin, Jacob (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Dong, Thi Ngan et al. (2021). "A multitask transfer learning framework for the prediction of virus-human protein–protein interactions". In: *BMC bioinformatics* 22, pp. 1–24.

Drugan, Jonelle K et al. (1996). "Ras Interaction with Two Distinct Binding Domains in Raf-1 5 Be Required for Ras Transformation". In: *Journal of Biological Chemistry* 271.1, pp. 233–237.

Duarte, Jose M et al. (2012). "Protein interface classification by evolutionary analysis". In: *BMC bioinformatics* 13, pp. 1–16.

Edelsbrunner, Herbert, David Letscher, and Afra Zomorodian (2000). *Proceedings 41st annual symposium on foundations of computer science*.

Elnaggar, Ahmed et al. (2021). "Prottrans: Toward understanding the language of life through self-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 44.10, pp. 7112–7127.

Eslami Manoochehri, Hafez and Mehrdad Nourani (2020). "Drug-target interaction prediction using semi-bipartite graph model and deep learning". In: *BMC bioinformatics* 21, pp. 1–16.

Evans, Richard et al. (2021). "Protein complex prediction with AlphaFold-Multimer". In: *biorxiv*, pp. 2021–10.

Feng, Ting et al. (2017). "HawkRank: a new scoring function for protein–protein docking based on weighted energy terms". In: *Journal of cheminformatics* 9, pp. 1–15.

Fu, Xiang et al. (2022). "Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations". In: *arXiv preprint arXiv:2210.07237*.

Fu, Xinyu et al. (2020). "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding". In: *Proceedings of the web conference 2020*, pp. 2331–2341.

Ganea, Octavian, Gary Bécigneul, and Thomas Hofmann (2018). "Hyperbolic neural networks". In: *Advances in neural information processing systems* 31.

Ganea, Octavian-Eugen et al. (2021). "Independent se (3)-equivariant models for end-to-end rigid protein docking". In: *arXiv preprint arXiv:2111.07786*.

Geng, Cunliang et al. (2019). "iSEE: Interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations". In: *Proteins: Structure, Function, and Bioinformatics* 87.2, pp. 110–119.

Genheden, Samuel and Ulf Ryde (2015). "The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities". In: *Expert opinion on drug discovery* 10.5, pp. 449–461.

Genheden, Samuel et al. (2012). "The normal-mode entropy in the MM/GBSA method: effect of system truncation, buffer region, and dielectric constant". In: *Journal of chemical information and modeling* 52.8, pp. 2079–2088.

Gilmer, Justin et al. (2017). "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR, pp. 1263–1272.

Gromov, Mikhael (1987). "Hyperbolic groups". In: *Essays in group theory*. Springer, pp. 75–263.

Grover, Aditya and Jure Leskovec (2016). "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864.

Gu, Xiang, Jian Sun, and Zongben Xu (2020). "Spherical space domain adaptation with robust pseudo-label loss". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9101–9110.

Hamilton, William L (2020). *Graph representation learning*. Morgan & Claypool Publishers.

Han, Jiaqi et al. (2022). "Geometrically equivariant graph neural networks: A survey". In: *arXiv preprint arXiv:2202.07230*.

Harary, Frank and Robert Z Norman (1960). "Some properties of line digraphs". In: *Rendiconti del circolo matematico di palermo* 9, pp. 161–168.

He, Neil, Menglin Yang, and Rex Ying (2024). "Lorentzian residual neural networks". In: *arXiv preprint arXiv:2412.14695*.

He, Xiangnan et al. (2017). "Neural collaborative filtering". In: *Proceedings of the 26th international conference on world wide web*, pp. 173–182.

Hiew, Shu Hui et al. (2023). "Modulation of mechanical properties of short bioinspired peptide materials by single amino-acid mutations". In: *Journal of the American Chemical Society* 145.6, pp. 3382–3393.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Hu, Weihua et al. (2019). "Strategies for pre-training graph neural networks". In: *arXiv preprint arXiv:1905.12265*.

Huang, Jing et al. (2017). "CHARMM36m: an improved force field for folded and intrinsically disordered proteins". In: *Nature methods* 14.1, pp. 71–73.

Huang, Kexin et al. (2020). "Caster: Predicting drug interactions with chemical substructure representation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 01, pp. 702–709.

Huang, Wenbing et al. (2022). "Equivariant graph mechanics networks with constraints". In: *arXiv preprint arXiv:2203.06442*.

Hussain, Syed Arshad (2009). "An introduction to fluorescence resonance energy transfer (FRET)". In: *arXiv preprint arXiv:0908.1815*.

Jankauskaitė, Justina et al. (2019). "SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation". In: *Bioinformatics* 35.3, pp. 462–469.

Javanainen, Matti, Hector Martinez-Seara, and Ilpo Vattulainen (2017). "Excessive aggregation of membrane proteins in the Martini model". In: *PloS one* 12.11, e0187936.

Jia, Guanbo et al. (2012). "Community detection in social and biological networks using differential evolution". In: *Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers*. Springer, pp. 71–85.

Jia, Jia et al. (2009). "Mechanisms of drug combinations: interaction and network perspectives". In: *Nature reviews Drug discovery* 8.2, pp. 111–128.

Jiang, Peiran et al. (2020). "Deep graph embedding for prioritizing synergistic anticancer drug combinations". In: *Computational and structural biotechnology journal* 18, pp. 427–438.

Jiang, Yelu et al. (2023). "DGCddG: deep graph convolution for predicting protein-protein binding affinity changes upon mutations". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 20.3, pp. 2089–2100.

Jiao, Zhenyu, Shuqi Sun, and Ke Sun (2018). "Chinese lexical analysis with deep bi-gru-crf network". In: *arXiv preprint arXiv:1807.01882*.

Jing, Bowen et al. (2020). "Learning from protein structure with geometric vector perceptrons". In: *International Conference on Learning Representations*.

Jing, Bowen et al. (2021). "Equivariant graph neural networks for 3d macromolecular structure". In: *arXiv preprint arXiv:2106.03843*.

Jong, Djurre H de, Xavier Periole, and Siewert J Marrink (2012). "Dimerization of amino acid side chains: lessons from the comparison of different force fields". In: *Journal of Chemical Theory and Computation* 8.3, pp. 1003–1014.

Jumper, John et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *nature* 596.7873, pp. 583–589.

Kabsch, Wolfgang and Christian Sander (1983). "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". In: *Biopolymers: Original Research on Biomolecules* 22.12, pp. 2577–2637.

Kamisetty, Hetunandan et al. (2011). "Accounting for conformational entropy in predicting binding free energies of protein-protein interactions". In: *Proteins: Structure, Function, and Bioinformatics* 79.2, pp. 444–462.

Kaufman, Shachar et al. (2012). "Leakage in data mining: Formulation, detection, and avoidance". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.4, pp. 1–21.

Keeler, James (2010). *Understanding NMR spectroscopy*. John Wiley & Sons.

Kingma, Diederik P (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kipf, Thomas et al. (2018). "Neural relational inference for interacting systems". In: *International conference on machine learning*. PMLR, pp. 2688–2697.

Kipf, Thomas N and Max Welling (2016). "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907*.

Kmiecik, Sebastian et al. (2016). "Coarse-grained protein models and their applications". In: *Chemical reviews* 116.14, pp. 7898–7936.

Kochurov, Max, Rasul Karimov, and Serge Kozlukov (2020). "Geoopt: Riemannian optimization in pytorch". In: *arXiv preprint arXiv:2005.02819*.

Koegl, Manfred and Peter Uetz (2007). "Improving yeast two-hybrid screening systems". In: *Briefings in Functional Genomics and Proteomics* 6.4, pp. 302–312.

Kong, Xiangzhe, Wenbing Huang, and Yang Liu (2022). "Conditional antibody design as 3d equivariant graph translation". In: *arXiv preprint arXiv:2208.06073*.

Kortemme, Tanja, Alexandre V Morozov, and David Baker (2003). "An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes". In: *Journal of molecular biology* 326.4, pp. 1239–1259.

Kroon, Peter C et al. (2022). "Martinize2 and vermouth: Unified framework for topology generation". In: *arXiv preprint arXiv:2212.01191*.

Kuhn, Michael et al. (2010). "A side effect resource to capture phenotypic effects of drugs". In: *Molecular systems biology* 6.1, p. 343.

Ladd, Marcus Frederick Charles, Rex Alfred Palmer, and Rex Alfred Palmer (1977). *Structure determination by X-ray crystallography*. Vol. 233. Springer.

Lamprakis, Christos et al. (2021). "Evaluating the efficiency of the Martini force field to study protein dimerization in aqueous and membrane environments". In: *Journal of Chemical Theory and Computation* 17.5, pp. 3088–3102.

Landrum, Greg et al. (2013). "RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling". In: *Greg Landrum* 8.31.10, p. 5281.

Lane, Thomas J et al. (2011). "Markov state model reveals folding and functional dynamics in ultra-long MD trajectories". In: *Journal of the American Chemical Society* 133.45, pp. 18413–18419.

Law, Marc et al. (2019). "Lorentzian distance learning for hyperbolic representations". In: *International Conference on Machine Learning*. PMLR, pp. 3672–3681.

Lelimousin, Mickaël, Vittorio Limongelli, and Mark SP Sansom (2016). "Conformational changes in the epidermal growth factor receptor: role of the transmembrane domain investigated by coarse-grained metadynamics free energy calculations". In: *Journal of the American Chemical Society* 138.33, pp. 10611–10622.

Levy, Emmanuel D et al. (2006). "3D complex: a structural classification of protein complexes". In: *PLoS computational biology* 2.11, e155.

Li, Minghui et al. (2016). "MutaBind estimates and interprets the effects of sequence variants on protein–protein interactions". In: *Nucleic acids research* 44.W1, W494–W501.

Lin, Jer-Sheng and Erh-Min Lai (2017). "Protein–protein interactions: co-immunoprecipitation". In: *Bacterial protein secretion systems: Methods and protocols*, pp. 211–219.

Lin, Zeming et al. (2023). "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637, pp. 1123–1130.

Lindsay, Mark A (2003). "Target discovery". In: *Nature Reviews Drug Discovery* 2.10, pp. 831–838.

Liu, Qiao and Lei Xie (2021). "TranSynergy: Mechanism-driven interpretable deep neural network for the synergistic prediction and pathway deconvolution of drug combinations". In: *PLoS computational biology* 17.2, e1008653.

Liu, Rui et al. (2011). "Meta-analysis of the relationship between Parkinson disease and melanoma". In: *Neurology* 76.23, pp. 2002–2009.

Liu, Sen et al. (2007). "Nonnatural protein–protein interaction-pair design by key residues grafting". In: *Proceedings of the National Academy of Sciences* 104.13, pp. 5330–5335.

Liu, Xiang et al. (2023). "Persistent tor-algebra for protein–protein interaction analysis". In: *Briefings in Bioinformatics* 24.2, bbad046.

Liu, Xianggen et al. (2021). "Deep geometric representations for modeling effects of mutations on protein-protein binding affinity". In: *PLoS computational biology* 17.8, e1009284.

Liu, Yufeng et al. (2022). "Rotamer-free protein sequence design based on deep learning and self-consistency". In: *Nature Computational Science* 2.7, pp. 451–462.

Loewe, S (1953). "The problem of synergism and antagonism of combined drugs". In: *Arzneimittel-forschung* 3.6, pp. 285–290.

Louche, Arthur, Suzana P Salcedo, and Sarah Bigot (2017). "Protein–protein interactions: pull-down assays". In: *Bacterial Protein secretion systems: methods and protocols*, pp. 247–255.

Luo, Yunan et al. (2017). "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information". In: *Nature communications* 8.1, p. 573.

Marrink, Siewert J and D Peter Tieleman (2013). "Perspective on the Martini model". In: *Chemical Society Reviews* 42.16, pp. 6801–6822.

Marrink, Siewert J et al. (2007). "The MARTINI force field: coarse grained model for biomolecular simulations". In: *The journal of physical chemistry B* 111.27, pp. 7812–7824.

Marrink, Siewert J et al. (2023). "Two decades of Martini: Better beads, broader scope". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 13.1, e1620.

McDonald, David (2021). "The hierarchical organisation and dynamics of complex networks". PhD thesis. University of Birmingham.

McDonald, David and Shan He (2022). "Hyperbolic embedding of attributed and directed networks". In: *IEEE Transactions on Knowledge and Data Engineering* 35.7, pp. 7003–7015.

Mirdita, Milot et al. (2022). "ColabFold: making protein folding accessible to all". In: *Nature methods* 19.6, pp. 679–682.

Monticelli, Luca et al. (2008). "The MARTINI coarse-grained force field: extension to proteins". In: *Journal of chemical theory and computation* 4.5, pp. 819–834.

Moon, Changsung et al. (2021). "Learning Drug-Disease-Target Embedding (DDTE) from knowledge graphs to inform drug repurposing hypotheses". In: *Journal of biomedical informatics* 119, p. 103838.

Mosca, Roberto et al. (2014). "3did: a catalog of domain-based interactions of known three-dimensional structure". In: *Nucleic acids research* 42.D1, pp. D374–D379.

Murata, Kazuyoshi and Matthias Wolf (2018). "Cryo-electron microscopy for structural analysis of dynamic biological macromolecules". In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1862.2, pp. 324–334.

Murphy, Allan H (1996). "The Finley affair: A signal event in the history of forecast verification". In: *Weather and forecasting* 11.1, pp. 3–20.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Navarro, Carles, Maciej Majewski, and Gianni De Fabritiis (2023). "Top-down machine learning of coarse-grained protein force fields". In: *Journal of Chemical Theory and Computation* 19.21, pp. 7518–7526.

Nguyen, Dac-Trung et al. (2017). "Pharos: collating protein information to shed light on the druggable genome". In: *Nucleic acids research* 45.D1, pp. D995–D1002.

Nguyen, Thin et al. (2021). "GraphDTA: predicting drug–target binding affinity with graph neural networks". In: *Bioinformatics* 37.8, pp. 1140–1147.

Oprea, Tudor I et al. (2011). "Drug repurposing from an academic perspective". In: *Drug Discovery Today: Therapeutic Strategies* 8.3-4, pp. 61–69.

Paszke, Adam et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Peng, Jiajie et al. (2021). "An end-to-end heterogeneous graph representation learning-based framework for drug–target interaction prediction". In: *Briefings in bioinformatics* 22.5, bbaa430.

Peng, Wei et al. (2022). "Improving cancer driver gene identification using multitask learning on graph convolutional network". In: *Briefings in Bioinformatics* 23.1, bbab432.

Peng, Xingang et al. (2023). "Characterizing the interaction conformation between T-cell receptors and epitopes with deep learning". In: *Nature machine intelligence* 5.4, pp. 395–407.

Periole, Xavier et al. (2012). "Structural determinants of the supramolecular organization of G protein-coupled receptors in bilayers". In: *Journal of the American Chemical Society* 134.26, pp. 10959–10965.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710.

Petsko, Gregory A and Dagmar Ringe (2004). *Protein structure and function*. New Science Press.

Pires, Douglas EV and David B Ascher (2016). "mCSM-AB: a web server for predicting antibody–antigen affinity changes upon mutation with graph-based signatures". In: *Nucleic acids research* 44.W1, W469–W473.

Porta-Pardo, Eduard et al. (2015). "A pan-cancer catalogue of cancer driver protein interaction interfaces". In: *PLoS computational biology* 11.10, e1004518.

Prasad, TS Keshava et al. (2009). "Human protein reference database—2009 update". In: *Nucleic acids research* 37.Database issue, p. D767.

Preuer, Kristina et al. (2018). "DeepSynergy: predicting anti-cancer drug synergy with Deep Learning". In: *Bioinformatics* 34.9, pp. 1538–1546.

Purkayastha, Sukannya et al. (2019). "Drug-drug interactions prediction based on drug embedding and graph auto-encoder". In: *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, pp. 547–552.

Ramsundar, Bharath et al. (2019). *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. " O'Reilly Media, Inc."

Ratcliffe, John G (2006). "Hyperbolic n-Manifolds". In: *Foundations of Hyperbolic Manifolds*, pp. 508–599.

Ravasz, Erzsébet and Albert-László Barabási (2003). "Hierarchical organization in complex networks". In: *Physical review E* 67.2, p. 026112.

Réau, Manon et al. (2023). "DeepRank-GNN: a graph neural network framework to learn patterns in protein–protein interfaces". In: *Bioinformatics* 39.1, btac759.

Ren, Gao-Peng et al. (2023). "Force field-inspired molecular representation learning for property prediction". In: *Journal of Cheminformatics* 15.1, p. 17.

Reynolds, William F (1993). "Hyperbolic geometry on a hyperboloid". In: *The American mathematical monthly* 100.5, pp. 442–455.

Rodrigues, Carlos HM et al. (2019). "mCSM-PPI2: predicting the effects of mutations on protein–protein interactions". In: *Nucleic acids research* 47.W1, W338–W344.

Roel-Touris, Jorge et al. (2019). "Less is more: coarse-grained integrative modeling of large biomolecular assemblies with HADDOCK". In: *Journal of chemical theory and computation* 15.11, pp. 6358–6367.

Rogers, David and Mathew Hahn (2010). "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5, pp. 742–754.

Rose, George D (2021). "Reframing the protein folding problem: Entropy as organizer". In: *Biochemistry* 60.49, pp. 3753–3761.

Rudolph, Markus G and Ian A Wilson (2002). "The specificity of TCR/pMHC interaction". In: *Current opinion in immunology* 14.1, pp. 52–65.

Ryu, Jae Yong, Hyun Uk Kim, and Sang Yup Lee (2018). "Deep learning improves prediction of drug–drug and drug–food interactions". In: *Proceedings of the national academy of sciences* 115.18, E4304–E4311.

Salomon-Ferrer, Romelia, David A Case, and Ross C Walker (2013). "An overview of the Amber biomolecular simulation package". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 3.2, pp. 198–210.

Santurkar, Shibani et al. (2018). "How does batch normalization help optimization?" In: *Advances in neural information processing systems* 31.

Satorras, Vıctor Garcia, Emiel Hoogeboom, and Max Welling (2021). "E (n) equivariant graph neural networks". In: *International conference on machine learning*. PMLR, pp. 9323–9332.

Schwede, Torsten et al. (2003). "SWISS-MODEL: an automated protein homology-modeling server". In: *Nucleic acids research* 31.13, pp. 3381–3385.

Schymkowitz, Joost et al. (2005). "The FoldX web server: an online force field". In: *Nucleic acids research* 33.suppl_2, W382–W388.

Sengupta, Durba and Siewert J Marrink (2010). "Lipid-mediated interactions tune the association of glycophorin A helix and its disruptive mutants in membranes". In: *Physical Chemistry Chemical Physics* 12.40, pp. 12987–12996.

Singh, Rohit et al. (2022). "Topsy-Turvy: integrating a global view into sequence-based PPI prediction". In: *Bioinformatics* 38.Supplement_1, pp. i264–i272.

Sirin, Sarah et al. (2016). "AB-bind: antibody binding mutational database for computational affinity predictions". In: *Protein Science* 25.2, pp. 393–409.

Smith, Micholas Dean et al. (2015). "Force-field induced bias in the structure of Aβ21–30: A comparison of OPLS, AMBER, CHARMM, and GROMOS force fields". In: *Journal of chemical information and modeling* 55.12, pp. 2587–2595.

Souza, Paulo CT et al. (2021). "Martini 3: a general purpose force field for coarse-grained molecular dynamics". In: *Nature methods* 18.4, pp. 382–388.

Stärk, Hannes et al. (2022). "Equibind: Geometric deep learning for drug binding structure prediction". In: *International conference on machine learning*. PMLR, pp. 20503–20521.

Su, Jin et al. (2023). "Saprot: Protein language modeling with structure-aware vocabulary". In: *bioRxiv*, pp. 2023–10.

Su, Minyi et al. (2018). "Comparative assessment of scoring functions: the CASF-2016 update". In: *Journal of chemical information and modeling* 59.2, pp. 895–913.

Sun, Li et al. (2021). "Hyperbolic variational graph neural network for modeling dynamic graphs". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 5, pp. 4375–4383.

Sun, Wei, Philip E Sanderson, and Wei Zheng (2016). "Drug combination therapy increases successful drug repositioning". In: *Drug discovery today* 21.7, pp. 1189–1195.

Sun, Yizhou et al. (2011). "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks". In: *Proceedings of the VLDB Endowment* 4.11, pp. 992–1003.

Szklarczyk, Damian et al. (2015). "STRING v10: protein–protein interaction networks, integrated over the tree of life". In: *Nucleic acids research* 43.D1, pp. D447–D452.

Talloen, Willem et al. (2007). "I/NI-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data". In: *Bioinformatics* 23.21, pp. 2897–2902.

Tan, Xu et al. (2012). "Systematic identification of synergistic drug pairs targeting HIV". In: *Nature biotechnology* 30.11, pp. 1125–1130.

Toutanova, Kristina and Danqi Chen (2015). "Observed versus latent features for knowledge base and text inference". In: *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pp. 57–66.

Townshend, Raphael JL et al. (2020). "Atom3d: Tasks on molecules in three dimensions". In: *arXiv preprint arXiv:2012.04035*.

Tsherniak, Aviad et al. (2017). "Defining a cancer dependency map". In: *Cell* 170.3, pp. 564–576.

Vaswani, A (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*.

Veličković, Petar et al. (2017). "Graph attention networks". In: *arXiv preprint arXiv:1710.10903*.

Walsh, Brian, Sameh K Mohamed, and Vít Nováček (2020). "Biokg: A knowledge graph for relational learning on biological data". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 3173–3180.

Wan, Fangping et al. (2019). "NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions". In: *Bioinformatics* 35.1, pp. 104–111.

Wan, Shunzhou et al. (2022). "Ensemble simulations and experimental free energy distributions: Evaluation and characterization of isoxazole amides as SMYD3 inhibitors". In: *Journal of Chemical Information and Modeling* 62.10, pp. 2561–2570.

Wang, Ercheng et al. (2019a). "End-point binding free energy calculation with MM/PBSA and MM/GBSA: strategies and applications in drug design". In: *Chemical reviews* 119.16, pp. 9478–9508.

Wang, Haiying et al. (2021). "Deep learning in systems medicine". In: *Briefings in Bioinformatics* 22.2, pp. 1543–1559.

Wang, Hongzhun et al. (2022a). "A heterogeneous network-based method with attentive meta-path extraction for predicting drug–target interactions". In: *Briefings in Bioinformatics* 23.4, bbac184.

Wang, Jinxian et al. (2022b). "DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations". In: *Briefings in Bioinformatics* 23.1, bbab390.

Wang, Menglun, Zixuan Cang, and Guo-Wei Wei (2020). "A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation". In: *Nature Machine Intelligence* 2.2, pp. 116–123.

Wang, Renxiao et al. (2004). "The PDBbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures". In: *Journal of medicinal chemistry* 47.12, pp. 2977–2980.

Wang, Renxiao et al. (2005). "The PDBbind database: methodologies and updates". In: *Journal of medicinal chemistry* 48.12, pp. 4111–4119.

Wang, Wujie (2022). "Generative Coarse-Graining". In: *APS March Meeting Abstracts*. Vol. 2022, N49–010.

Wang, Wujie and Rafael Gómez-Bombarelli (2019). "Coarse-graining auto-encoders for molecular dynamics". In: *npj Computational Materials* 5.1, p. 125.

Wang, Xiao et al. (2019b). "Heterogeneous graph attention network". In: *The world wide web conference*, pp. 2022–2032.

Wang, Zichen et al. (2024). "EGGNet, a generalizable geometric deep learning framework for protein complex pose scoring". In: *ACS omega* 9.7, pp. 7471–7479.

Webb, Benjamin and Andrej Sali (2016). "Comparative protein structure modeling using MODELLER". In: *Current protocols in bioinformatics* 54.1, pp. 5–6.

Wee, JunJie and Kelin Xia (2022). "Persistent spectral based ensemble learning (PerSpect-EL) for protein–protein binding affinity prediction". In: *Briefings in Bioinformatics* 23.2, bbac024.

Whisstock, James C and Arthur M Lesk (2003). "Prediction of protein function from protein sequence and structure". In: *Quarterly reviews of biophysics* 36.3, pp. 307–340.

Wishart, David S et al. (2006). "DrugBank: a comprehensive resource for in silico drug discovery and exploration". In: *Nucleic acids research* 34.suppl_1, pp. D668–D672.

Wu, Fang et al. (2022a). "Pre-Training of Equivariant Graph Matching Networks with Conformation Flexibility for Drug Binding". In: *Advanced Science* 9.33, p. 2203796.

Wu, Shiwen et al. (2022b). "Graph neural networks in recommender systems: a survey". In: *ACM Computing Surveys* 55.5, pp. 1–37.

Wu, Zhenxing et al. (2021). "Hyperbolic relational graph convolution networks plus: a simple but highly efficient QSAR-modeling method". In: *Briefings in Bioinformatics* 22.5, bbab112.

Xiong, Peng et al. (2017). "BindProfX: assessing mutation-induced binding affinity change by protein interface profiles with pseudo-counts". In: *Journal of molecular biology* 429.3, pp. 426–434.

Xu, Keyulu et al. (2018). "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826*.

Xu, Minkai et al. (2022). "Geodiff: A geometric diffusion model for molecular conformation generation". In: *arXiv preprint arXiv:2203.02923*.

Xu, Qifang and Roland L Dunbrack (2010). "The protein common interface database (ProtCID)—a comprehensive database of interactions of homologous proteins in multiple crystal forms". In: *Nucleic acids research* 39.suppl_1, pp. D761–D770.

Yadav, Bhagwan et al. (2015). "Searching for drug synergy in complex dose–response landscapes using an interaction potency model". In: *Computational and structural biotechnology journal* 13, pp. 504–513.

Yang, Bishan et al. (2014). "Embedding entities and relations for learning and inference in knowledge bases". In: *arXiv preprint arXiv:1412.6575*.

Yang, Menglin et al. (2022). "Hyperbolic graph neural networks: A review of methods and applications". In: *arXiv preprint arXiv:2202.13852*.

Ye, Qing et al. (2021). "A unified drug–target interaction prediction framework based on knowledge graph and recommendation system". In: *Nature communications* 12.1, p. 6775.

Yellaboina, Sailu et al. (2011). "DOMINE: a comprehensive collection of known and predicted domain-domain interactions". In: *Nucleic acids research* 39.suppl_1, pp. D730–D735.

Yu, Ke, Shyam Visweswaran, and Kayhan Batmanghelich (2020). "Semi-supervised hierarchical drug embedding in hyperbolic space". In: *Journal of chemical information and modeling* 60.12, pp. 5647–5657.

Yu, Yue et al. (2021a). "SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization". In: *Bioinformatics* 37.18, pp. 2988–2995.

Yu, Zhouxin et al. (2021b). "Predicting drug–disease associations through layer attention graph convolutional network". In: *Briefings in bioinformatics* 22.4, bbaa243.

Yue, Yang et al. (2023a). "FLONE: fully Lorentz network embedding for inferring novel drug targets". In: *Bioinformatics Advances* 3.1, vbad066.

Yue, Yang et al. (2023b). "Improving therapeutic synergy score predictions with adverse effects using multi-task heterogeneous network learning". In: *Briefings in Bioinformatics* 24.1, bbac564.

Yue, Yang et al. (2023c). "MpbPPI: a multi-task pre-training-based equivariant approach for the prediction of the effect of amino acid mutations on protein–protein interactions". In: *Briefings in Bioinformatics* 24.5, bbad310.

Yue, Yang et al. (2024). "Integration of molecular coarse-grained model into geometric representation learning framework for protein-protein complex property prediction". In: *Nature Communications* 15.1, p. 9629.

Zeng, Yuansong et al. (2023). "Identifying B-cell epitopes using AlphaFold2 predicted structures and pretrained language model". In: *Bioinformatics* 39.4, btad187.

Zhang, Daoqiang, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. (2012). "Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease". In: *NeuroImage* 59.2, pp. 895–907.

Zhang, Li (2022). "Exploring local information for graph representation learning". PhD thesis. University of Sheffield.

Zhang, Ping et al. (2013). "Exploring the relationship between drug side-effects and therapeutic indications". In: *AMIA annual symposium proceedings*. Vol. 2013. American Medical Informatics Association, p. 1568.

Zhang, Yang and Jeffrey Skolnick (2005). "TM-align: a protein structure alignment algorithm based on the TM-score". In: *Nucleic acids research* 33.7, pp. 2302–2309.

Zhang, Yiding et al. (2021). "Lorentzian graph convolutional networks". In: *Proceedings of the web conference 2021*, pp. 1249–1261.

Zhang, Yu and Qiang Yang (2021). "A survey on multi-task learning". In: *IEEE transactions on knowledge and data engineering* 34.12, pp. 5586–5609.

Zhang, Zuobai et al. (2022). "Protein representation learning by geometric structure pretraining". In: *arXiv preprint arXiv:2203.06125*.

Zhang, Zuobai et al. (2024). "Pre-training protein encoder via siamese sequence-structure diffusion trajectory prediction". In: *Advances in Neural Information Processing Systems* 36.

Zheng, Shuyu et al. (2021). "DrugComb update: a more comprehensive drug sensitivity data repository and analysis portal". In: *Nucleic acids research* 49.W1, W174–W184.

Zhou, Bingxin et al. (2024). "Protein engineering with lightweight graph denoising neural networks". In: *Journal of Chemical Information and Modeling* 64.9, pp. 3650–3661.

Zhou, Huan-Xiang and Sanbo Qin (2007). "Interaction-site prediction for protein complexes: a critical assessment". In: *Bioinformatics* 23.17, pp. 2203–2209.

Zhu, Zhaocheng et al. (2022). "Torchdrug: A powerful and flexible machine learning platform for drug discovery". In: *arXiv preprint arXiv:2202.08320*.

Zimprich, Alexander et al. (2004). "Mutations in LRRK2 cause autosomal-dominant parkinsonism with pleomorphic pathology". In: *Neuron* 44.4, pp. 601–607.

Zitnik, Marinka, Monica Agrawal, and Jure Leskovec (2018). "Modeling polypharmacy side effects with graph convolutional networks". In: *Bioinformatics* 34.13, pp. i457–i466.