

# Graphical Neural Networks

Dominic DiSanto

Department of Biostatistics, Harvard University

December 6, 2023

Downloadable Slides



# Outline

- 1 Set-Up and Motivation
- 2 General Construction
- 3 Applications and Extensions
  - Structure-Based Drug Design (SBDD)
  - Multimodal Graph Learning (MGL)
  - TxGNN

# Goals

- Briefly motivate and introduce Graphical Neural Networks (GNN)
- Describe recent reviews and applications
  - Drug Discovery
  - Multimodal Biomedical Data
  - Knowledge Graph Integrated GNN's & TxGNN

# Knowledge Graph Data

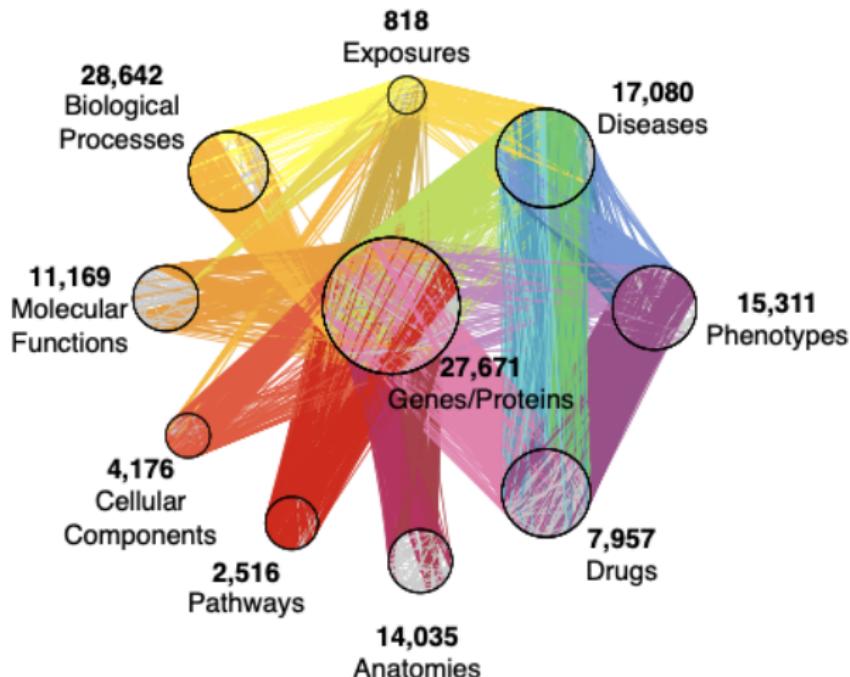


Figure: Fig 1A from Huang (2023) [9]

Downloadable Slides

# Multi-modal Biomedical Data

Multimodal knowledge graph  
of 17,080 disease phenotypes

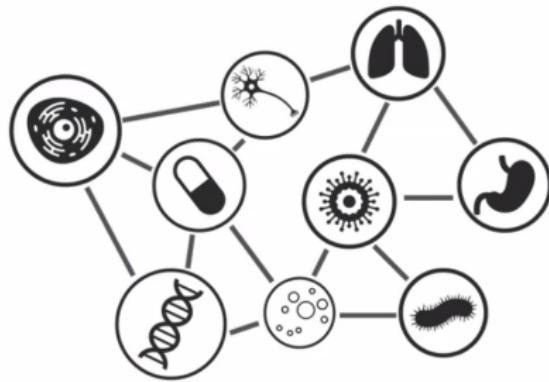
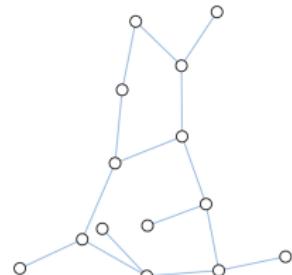
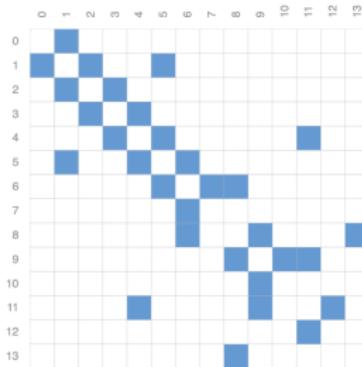
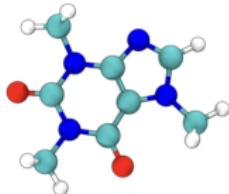


Image courtesy of partial figure from McDermott et al. *Structure-inducing pre-training* [14]

Downloadable Slides

# Molecular/Biochemical



(Left) 3d representation of the Caffeine molecule (Center) Adjacency matrix of the bonds in the molecule (Right) Graph representation of the molecule.

Image courtesy of <https://distill.pub/2021/gnn-intro/> [16]

# Protein Representation

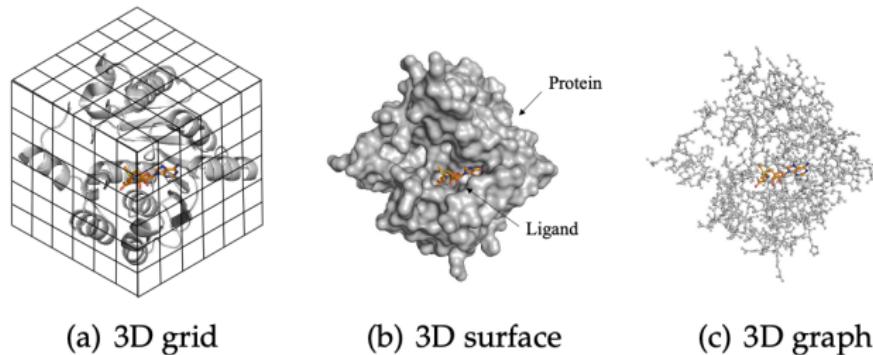


Fig. 2. 3D representations of proteins used for geometric deep learning: (a) 3D grid, (b) 3D surface, and (c) 3D graph, illustrated for PDB ID 2avd.

Fig 2. of Zhang (2023) *Geometric Deep Learning for Structure-Based Drug Design: A Survey* [21]

# Motivation

- Want to utilize the input structure of the graph
  - Respect/Maintain
  - Update/Estimate
- "Flattening" graphical data for DNN, CNN, etc. omits useful topology from our data
- Early methods attempting to retain topological info included recursive neural networks and random walk models, which GNN methodology extended [17]

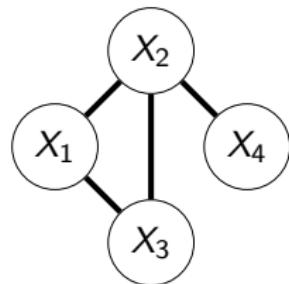
# Notation/Set-Up

- Consider the graph  $\mathcal{G} = (V, E)$ ,  $E \subseteq V \times V$ , where any node  $v$  has a related "feature vector"  $x_v \in \mathbb{R}^d$ 
  - Let  $N = |V|$

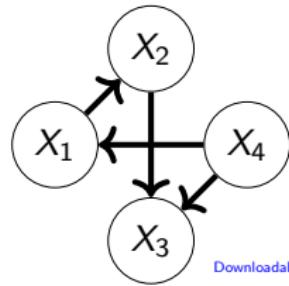
- Let  $\mathcal{N}_s(v)$  represent the  $s$ -hop neighborhood of any node  $v$  (and implicitly  $\mathcal{N}(v) \equiv \mathcal{N}_1(v)$ )

- Can construct adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  to capture structure of edge set  $E$ 
  - $\mathbf{A}_{ij} = w_{ij} \mathbb{1}\{(i,j) \in E\}$  for scalar weight  $w_{ij} \in \mathbb{R}$

Undirected Graph



Directed Graph



Downloadable Slides

# Topology Representations

- Simplest/Naïve method is to use  $\mathbf{A}$
- Consider also the Laplacian matrix  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ 
  - $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}_N)$
- Can use an eigenvalue-normalized Laplacian
$$\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$$
- Note that a given graph topology is represented equivalently by any permutation of its  $\mathbf{A}$

# Outline

1 Set-Up and Motivation

2 General Construction

3 Applications and Extensions

- Structure-Based Drug Design (SBDD)
- Multimodal Graph Learning (MGL)
- TxGNN

Downloadable Slides



# What do we estimate about graph structure?

GNN learning can be  $\left\{ \begin{array}{l} \text{Node-wise } \Phi(\mathcal{G}, v) : (v \in V) \rightarrow \mathbb{R}^m \\ \text{Edge-wise } \Phi(\mathcal{G}, e) : (e \in E) \rightarrow \mathbb{R}^m \\ \text{Graph-level characteristics } \Phi(\mathcal{G}) \end{array} \right.$

# What do we estimate about graph structure?

GNN learning can be

$$\left\{ \begin{array}{l} \text{Node-wise } \Phi(\mathcal{G}, x) : (x \in V) \rightarrow \mathbb{R}^m \\ \text{Edge characteristics } \Phi(\mathcal{G}, e) : (e \in E) \rightarrow \mathbb{R}^m \\ \text{Graph-level characteristics } \Phi(\mathcal{G}) \end{array} \right.$$

# General Framework<sup>1</sup>

We begin with the general Message Passing Neural Networks (MPNN) structure of a GNN:

- 1: Initialize  $h^{(0)} \leftarrow x_v, \forall v \in V$
- 2: **for**  $\kappa = 0, \dots, K$  **do**:
- 3:   **for**  $v \in \mathcal{G}$  **do**:
- 4:      $m_u^{(\kappa+1)} \leftarrow \text{Message}(h_v^{(\kappa)}, h_u^{(\kappa)}, e_{vu}), \forall u \in \mathcal{N}_v$
- 5:      $h_{agg}^{(\kappa+1)} \leftarrow \text{Aggregate}(\{m_u^{(\kappa)} \mid u \in \mathcal{N}_v\})$
- 6:      $h^{(\kappa+1)} \leftarrow \text{Update}(h^{(\kappa)}, h_{agg}^{(\kappa)})$
- 7:      $\hat{y} \leftarrow \text{Transform}(\{h_v^K \mid v \in \mathcal{G}\})$                or  $\text{Readout}(\cdot)$

---

<sup>1</sup>See [4, 7, 19]

# General Framework

Can succinctly represent the  $\kappa$ th layer as:

$$h_v^{(\kappa+1)} = Up \left\{ h_v^{(\kappa)}, Agg \left[ Msg \left( h_v^{(\kappa)}, x_u^{(\kappa)}, e_{u,v}^{(\kappa)} \right) \right] \right\}$$

Choices of (differentiable) functions for Aggregate, Update, and Readout determine the architecture of your GNN

Trained end-to-end via backpropagation for problem-specific Transform function

# Aggregate & Update

- **Aggregate(·)** produces a representation of information from a node's neighborhood via **permutation invariant** function
- Can include weights (edge-wise or learned)
- Over later iterations, this includes information from further and further distant nodes to any one target node
- We then **Update(·)** our current state using this aggregated neighborhood-level information

# Transform/Readout

- **Transform(·)** translates our learned node representations to some desired outcome
  - Regression
  - Binary/Multi-class classification
  - MLP/DNN's
  - **Readout(·)** is common term for translating node-level output to graph-level
  - *Global Pooling* - Methods applied over entire graph (e.g. averaging, fitting "regular" deep neural network, etc.)

# Graph Convolutional Network

- Proposed in 2017 by Thomas Kipf, Max Welling [12], can consider one example of "Laplacian-based methods" [7]

$$\mathbf{H}^{(\kappa+1)} = \text{ReLU} \left( \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\kappa)} \Theta \right)$$

- Motivated by considering the graph convolution<sup>2</sup>  $x * g = U g U^T x$  as the message passing function
- Learned weight/parameter matrix  $\Theta$

---

<sup>2</sup>  $U$  the matrix of eigenvectors of  $\mathbf{L}$

# Graph Convolutional Network

Intuitive "derivation":

$$\mathbf{H}^{(\kappa+1)} = \sigma(\mathbf{A}\mathbf{H}^{(\kappa)}\Theta)$$

$$\mathbf{H}^{(\kappa+1)} = \sigma(\mathbf{D}^{-1}\mathbf{A}\mathbf{H}^{(\kappa)}\Theta) \quad \text{Normalizing by degree}$$

$$\mathbf{H}^{(\kappa+1)} = \sigma(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{H}^{(\kappa)}\Theta) \quad \text{Symmetric normalization}$$

$$\mathbf{H}^{(\kappa+1)} = \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(\kappa)}\Theta) \quad \text{Adding self-loop}$$

where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ ,  $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ ,  $\sigma$  is any activation function,  $\mathbf{H}$  is simply the matrix of  $\mathbf{h}_v$  for all nodes

# Graph Convolutional Network

- Proposed in 2017 by Thomas Kipf, Max Welling [12], can consider one example of "Laplacian-based methods" [7]

$$\mathbf{H}^{(\kappa+1)} = \underbrace{\text{ReLU}}_{\text{Update}} \underbrace{\left( \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\kappa)} \Theta \right)}_{\text{Aggregate/Message}}$$

- Motivated by considering the graph convolution<sup>3</sup>  $x * g = U g U^T x$  as the message passing function
- Learned weight/parameter matrix  $\Theta$

<sup>3</sup>  $U$  the matrix of eigenvectors of  $\mathbf{L}$

# General Framework (Revisited)

- **Input**

- Node conceptualization
- Node embeddings

- **Architecture**

- Message Passing Neural Network (MPNN)
- $\mathbf{H}^{(\kappa+1)} = \text{ReLU} \left( \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\kappa)} \Theta \right)$

- **Output**

- Target output (Readout, Transform functions)
- Loss function

# Outline

1 Set-Up and Motivation

2 General Construction

3 Applications and Extensions

- Structure-Based Drug Design (SBDD)
- Multimodal Graph Learning (MGL)
- TxGNN

Downloadable Slides

# Proteins and Structure-Based Drug Design (SBDD)

Geometric Deep Learning for Structure-Based Drug Design: A Survey  
(2023)

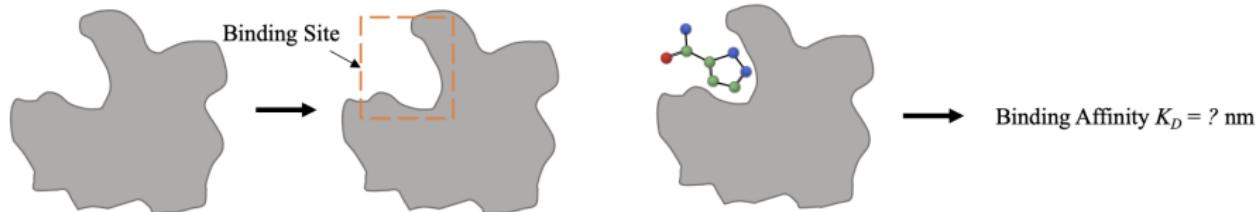
Zaixi Zhang, Jiaxian Yan, Qi Liu, Enhong Chen, and Marinka Zitnik [21]

# Structure-Based Drug Design (SBDD)

- SBDD aims to improve drug-discovery by understanding 3D protein structures and predicting drug efficacy/behavior
- Can dichotomize tasks categorized as *predictive* or *generative*
- Problems of note for GNN's include predictive tasks
  - **binding site prediction**
  - **binding affinity prediction**
- Other characteristics are important for drug design and protein-ligand interactions but without current GNN methods to my knowledge
  - Binding pose
  - Ligand generation
  - Linker design

# Output

- **Binding site prediction** is binary categorization at the (surface) amino acid level
- **Binding affinity prediction** is a continuous measure of protein-ligand interaction strength



Images from Fig. 1 of Zhang (2023) [21]

# Input

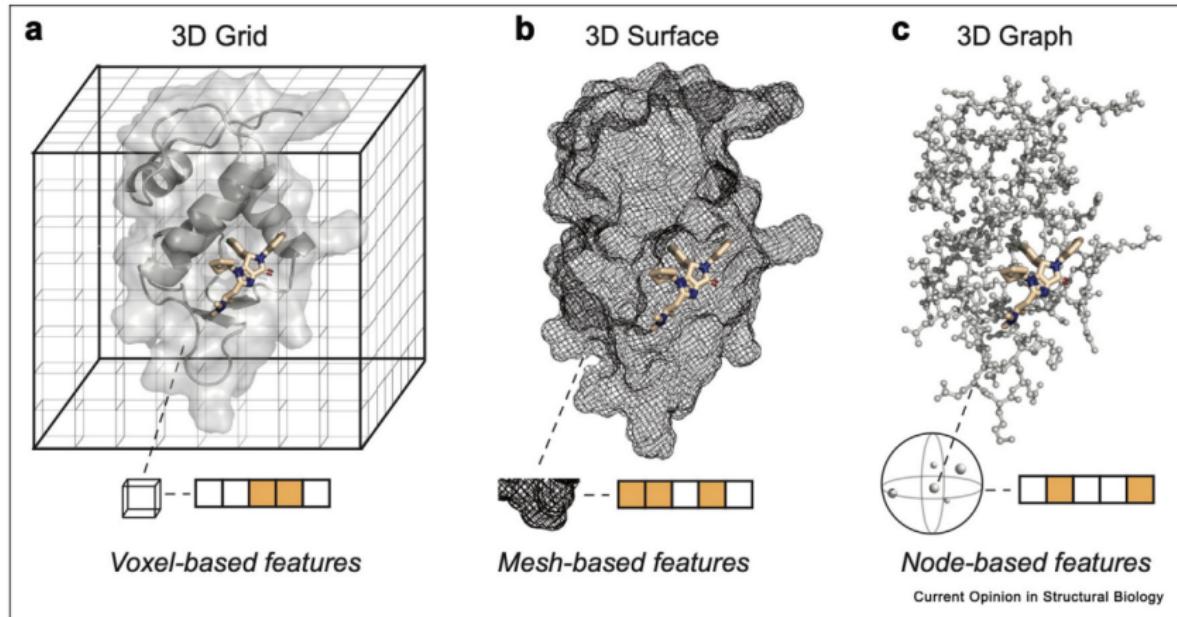
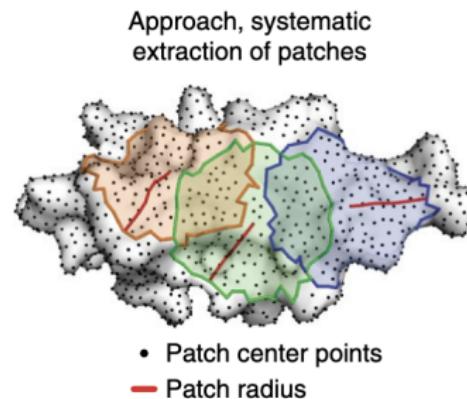
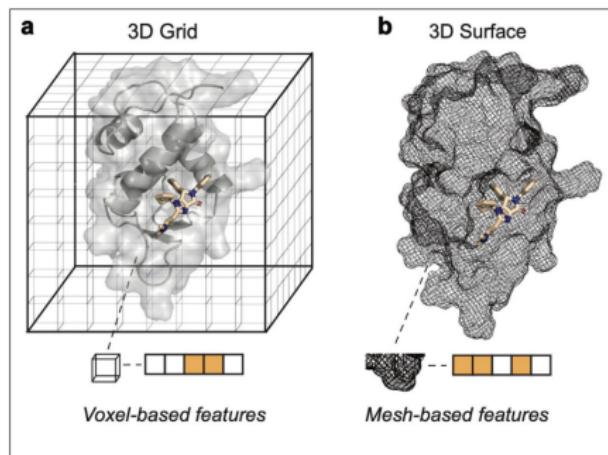


Figure from Isert (2023) [10]

Downloadable Slides

# CNN Input

- Nodes and edges defined by radial patches about a discretization of the protein surface

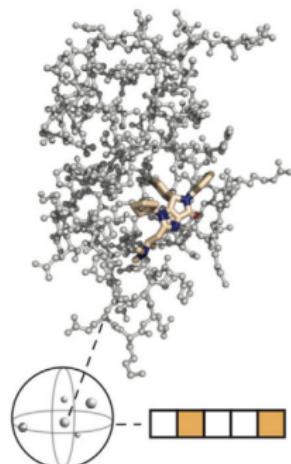


Left: Fig 2(ab) from Isert (2023); Right: Fig 1a. Gainza (2020) [10, 6]

# GNN Input

- Nodes  $v$  include typical feature embeddings  $h$  and 3-dimensional coordinate data  $\mathbf{v}$
- Nodes are primarily conceptualized at the **amino acid** (also **residue**) level
- Maintain invariance for translations, rotations but not necessarily reflections

C 3D Graph



*Node-based features*

Current Opinion in Structural Biology

Figure: Figure 1c from Isert (2023) [10]

Downloadable Slides

# Architecture

Consider some transformations  $T, T'$  (e.g. reflection, rotation, etc.) within the same symmetry group:

Invariance:  $f(T(x)) = T'(f(x)) = f(x)$

Equivariance:  $\forall T, \exists T' : f(T(x)) = T'(f(x))$

# Architecture(s)

Forgoing the iteration supercripts, one message pass can be represented as such:

$$m_{ij} = \text{Msg}_m(\mathbf{v}_i, \mathbf{v}_j, h_i, h_j, e_{ij})$$

$$\mathbf{m}_{ij} = \text{Msg}_m(\mathbf{v}_i, \mathbf{v}_j, h_i, h_j, e_{ij})$$

$$h'_i = \text{Update}_h [h_i, \text{Agg}_h(\{m_{ij}\}_{j \in \mathcal{N}(v_i)})]$$

$$\mathbf{v}'_i = \text{Update}_v [\mathbf{v}_i, \text{Agg}_v (\{\mathbf{m}_{ij}\}_{j \in \mathcal{N}(v_i)})]$$

where  $\text{Msg}_m$ ,  $\text{Update}_h$  are geometrically **invariant**, scalar functions and  $\text{Msg}_m$ ,  $\text{Update}_v$  are geometrically **equivariant**, vector functions  
Here  $\mathbf{v} \in \mathbb{R}^3$  are 3-D coordinates (e.g. atom or amino acid location)

# NodeCoder

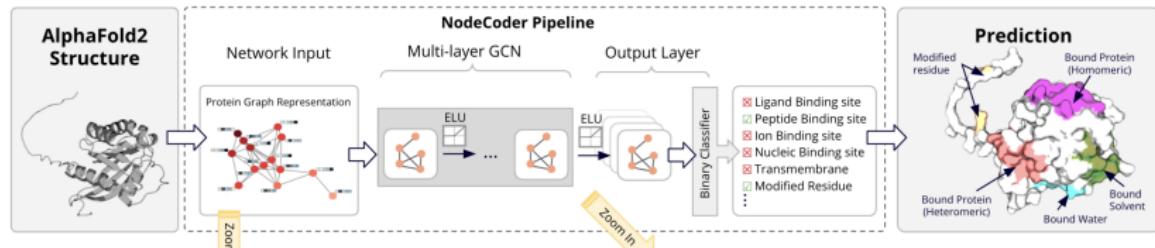


Figure: Subst of Fig. 1 from Abdollahi (2023) NodeCoder [1]

- Nodes are residues with atomic property (and other) embeddings
- With a given layer updated using a slight adaptation of GCN:

$$\mathbf{H}^{(\kappa+1)} = \text{ELU} \left( \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\kappa)} \Theta \right)$$

$$\mathbf{H}^{(K)} = \text{LogSoftMax} \left\{ \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \text{ELU} \left( \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\kappa)} \Theta \right) \Theta_t \right\}$$

Downloadable Slides

# PIGNet

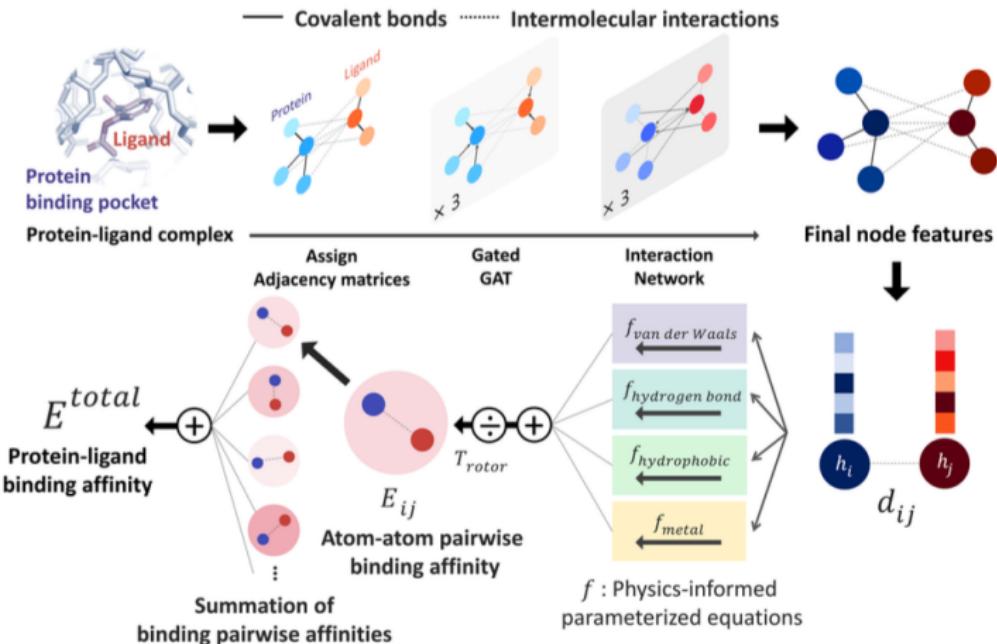


Figure: Fig 8. from Zhang (2023) describing PIGNet architecture [15, 21]

Downloadable Slides

# ScanNet

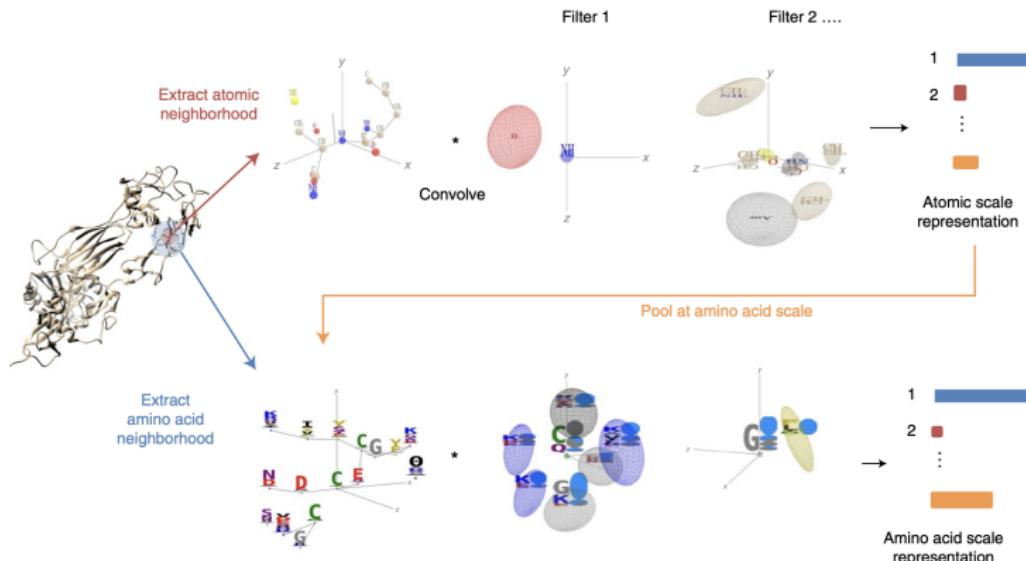


Figure: Fig 1. from Tubiana (2021) [18]

Downloadable Slides

# SBDD Summary

- Geometric Deep Learning is used to predict protein properties (in addition to drug-ligand interactions)
- Conceptualize nodes as residues
  - Can also fix graph with atomic nodes, fit GNN (or other method) to input atomic graph, and pool at the residue level
- Adapt our architecture to account for desired geometric invariance and equivariance
- Zhang (2023) [21] review compiles these and more complex architectures
  - Transformers
  - Variational Autoencoders
  - Methods for generative GDL tasks

Downloadable Slides

# Multimodal Graph Learning (MGL)

Multimodal learning with graphs (2023)

Yasha Ektefaie, George Dasoulas, Ayush Noori, Maha Farhat, and Marinka Zitnik

# Motivation

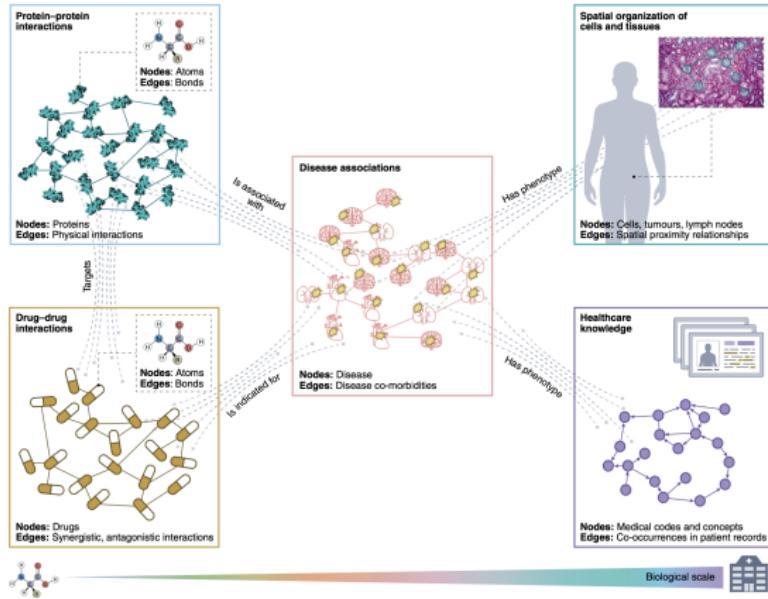


Figure: Fig. 3 from Li (2022) [13]

Downloadable Slides

# Multimodal Graph Learning (MGL)

- Ektefaie (2023) *Multimodal learning with graphs* [4]
- Topology is complicated by multimodality of input data
  - Modal collapse [11]
  - Differential data availability

## Clinical Data

- Clinical text
- -omics data
- Laboratory measurements
- Clinical imaging

## Protein Structures

- 1° AA Sequence
- 2° Helix interactions
- 3° Folding, bridges

# Aggregation

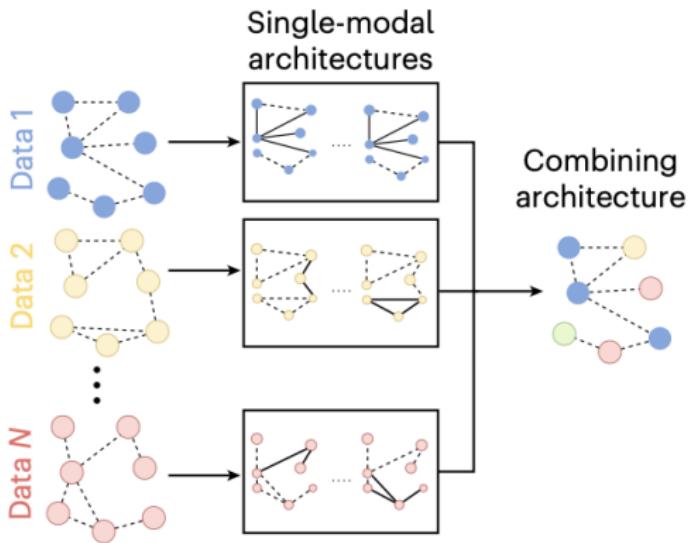


Figure: Subet of Fig. 2 from Ektefaie (2023) [4]

# Framework

Authors propose a four component "blueprint"

- ① Identifying entities (i.e. modalities)
- ② Uncovering topology
  - *A priori*
  - Adaptively learned
- ③ Propagating information
- ④ Mixing representation

# Framework

Authors propose a four component "blueprint"

- ① Identifying entities (i.e. modalities)      } Structure Learning
- ② Uncovering topology                          }
- ③ Propagating information                      } Learning On-Structure Phase
- ④ Mixing representation                        }

# Structure Learning

- Consider modalities as *entities* (colored nodes)
  - Clinical text/narrative data
  - Laboratory/Physiological measurements
  - Image/Video data
  - Patient reported measurements/symptoms
  - etc.
- Use known modalities to **identify** nodes
- **Connect** nodes to generate graph structure (learned or provided *a priori* knowledge)

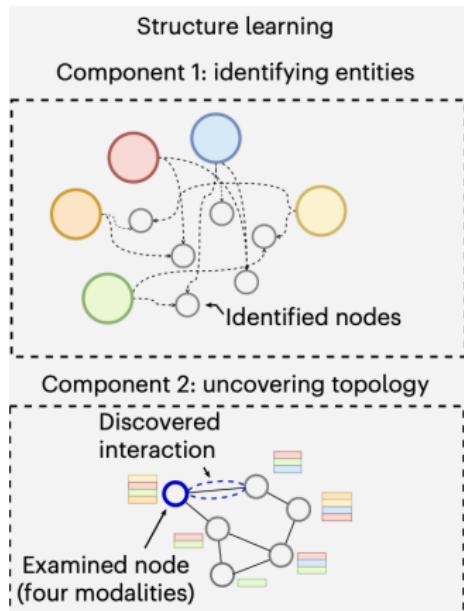


Figure: Subset of Fig 2c [4]

Downloadable Slides

# Learning on Structure

- Can construct one or more adjacency measures  $\mathbf{A}$
- Message propagate across edges outlined in Structure Learning
- Combine representations
  - Summation
  - Averaging
  - Neighborhood-specific aggregation

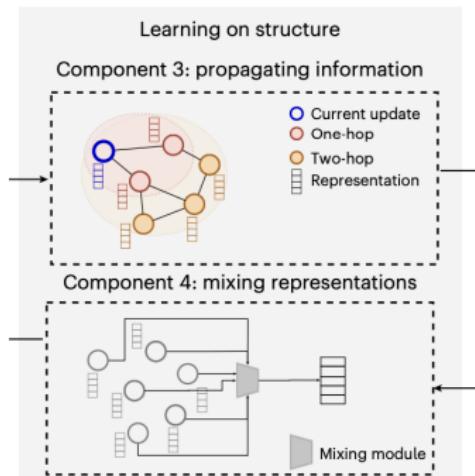


Figure: Subset of Fig 2c [4]

Method	Identifying entities	Uncovering topology	Propagating information	Mixing representations	Application
FuNet [21]	Hyperspectral pixels	Radial basis function similarity	minigCN (GCN mini-batching)	Concatenation, sum, or product	Hyperspectral image classification
Graph-FCN [22]	Pixels	Edge weights based on a Gaussian kernel	GCN on weighted edges	Graph loss added with fully connected network	Image semantic segmentation
GSMN [23]	Images, relations, and attributes	Visual graph for images combined with textual graph	Node-level and graph-level matching	Similarity function for positive and negative pairs	Image-text matching
RAG-GAT [24]	Super-pixels	Region adjacency graph	Graph attention network	Sum pooling combined with an MLP	Superpixel image classification
TextGCN [25]	Words and documents	Occurrence edges in text and corpus	GCN	No mixing, single-channel model	Text classification
CoGAN [26]	Sentences and aspects	Sentences and aspects as nodes	Cooperative graph attention	Softmax decoding block	Aspect sentiment classification
MCN [27]	Sentences, mentions, and entities	Document-level graph	Relation-aware GCN	Sigmoid activation on entity pairs	Document-level relation extraction
GP-GNN [28]	Word and position encodings	Generated adjacency Matrix	Neural message passing	Pair-wise product	Relation extraction
QM-GNN [29]	Atoms	Chemical bonds	Weisfeiler-Lehman network and global attention	Concatenation with quantum mechanical descriptors	Regio-selectivity prediction
GNS [30]	particles	Radial particle connectivity	Graph network (learned directed message passing)	No mixing, single-channel model	Particle-based simulation
MaSIF [31]	Discretized protein surface mesh	Overlapping geodesic radial features	Gaussian kernels on a local geodesic system	No mixing, single-channel model	Ligand site prediction and classification
MMGL [32]	Patients	Modality-aware latent graph	Adaptive GCN	Sub-branch prediction neural network	Disease prediction

Figure: Supp. Table 2 from Ektefaie (2023) [4]

# TextGCN

- Graph-building by word co-occurrence (*uncovering topology*)
- $Z = \text{SoftMax}(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\text{ReLU}(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(\kappa)}\Theta)\Theta^*)$

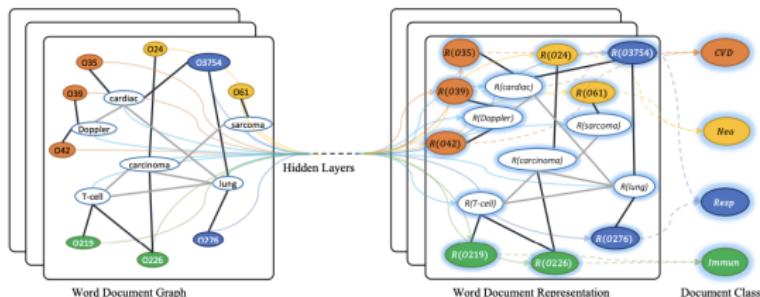
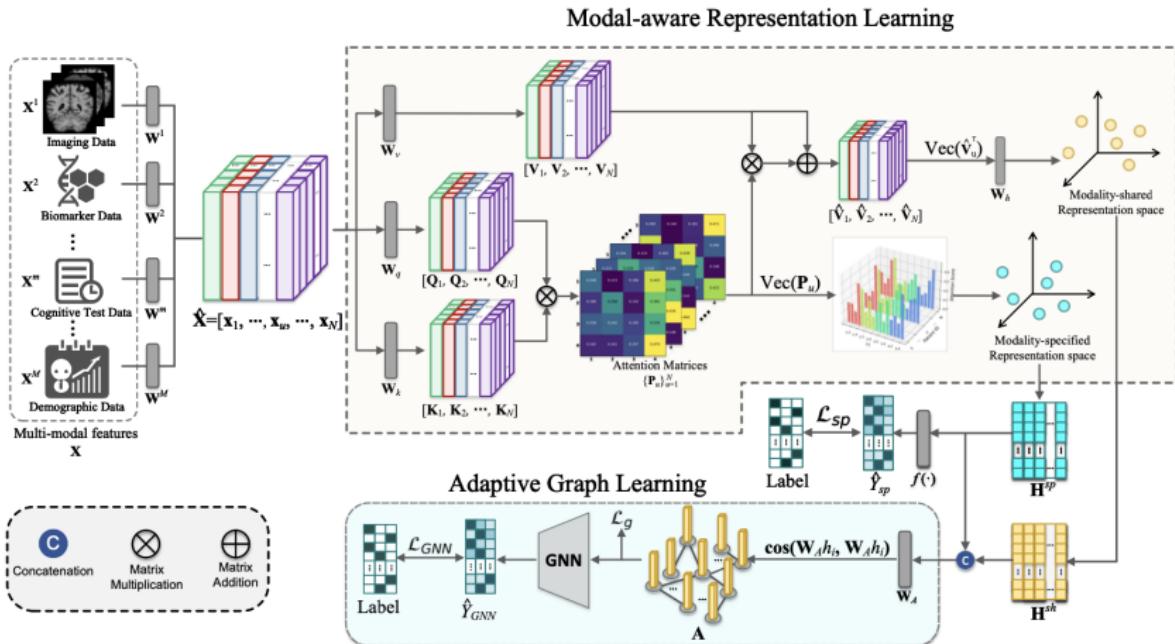


Figure: Figure 1 from Yao (2018) [20]

# Zheng (2022) Multi-modal Graph Learning for Disease Prediction



# Summary of Multimodal Graph Learning

- Accounting for intermodal information (currently) arises in graph-learning stages
- Quite comprehensive framework, includes some early GDL/CNN papers and methods
  - MaSIF CNN model
  - Graph Attention Networks

# TxGNN: Geometric Deep Learning and "Human-Centered AI"

Zero-shot prediction of therapeutic use with geometric deep learning and clinician centered design (2023)

Huang, Chandak, Wang, Havaldar, Vaid, Leskovec, Nadkarni, Glicksberg, Gehlenborg, Zitnik

# Motivation

- TxGNN trained to predict therapeutic agents for "neglected" diseases
- "Zero-shot prediction" - disease with few or no understood indications
- Formally stated, we want to predict if drug  $j$  is indicated or contraindicated for disease  $i$

# Input

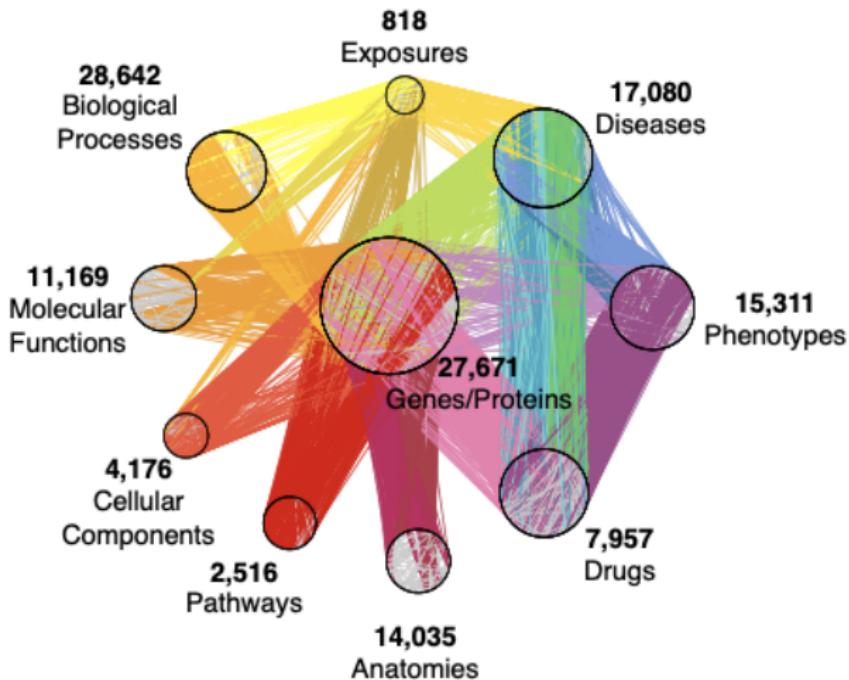


Figure: Fig 1A from Huang (2023) [9]

# Harmonization and PrimeKG

- Chandak (2023) "*Building a knowledge graph to enable precision medicine*" paper provides framework on KG harmonization [2]
- Define common ontologies and construct an aggregated KG from heterogeneous sources
  - "drug", "disease", "anatomy", "pathway", "biological process", "molecular function", "gene/protein", "effect/phenotype", "exposure"

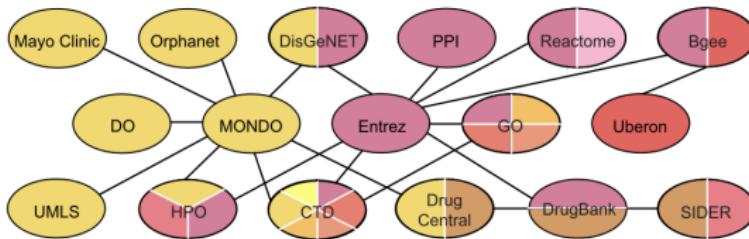


Figure: Fig. 2a from Chandak (2023) [2]<sup>4</sup>

<sup>4</sup>This does not reflect the specific data sources in TxGNN, although there is some overlap, just a general data harmonization outline

# Input

Represent our KG  $\mathcal{G} = (V, E, \mathcal{T}_R)$ ,  $e_{ij} = (v_i, r, v_j)$  and our usual node-embedding notation  $h_i$

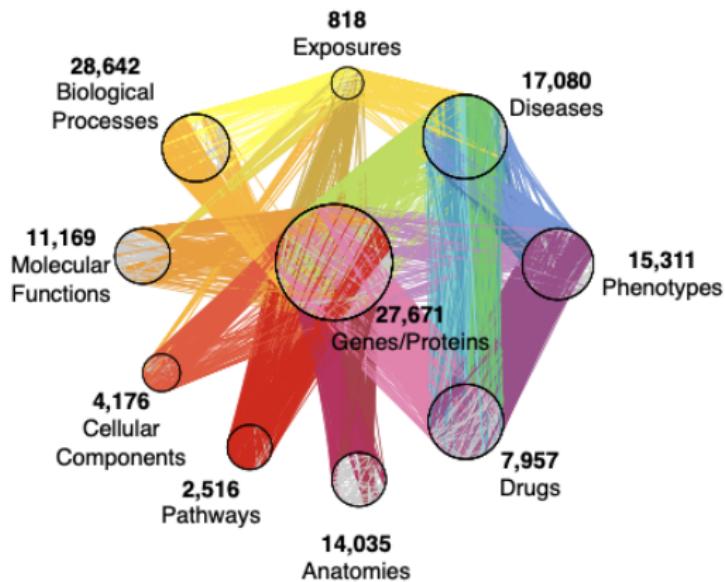


Figure: Fig 1A from Huang (2023) [9]

Downloadable Slides

# Architecture

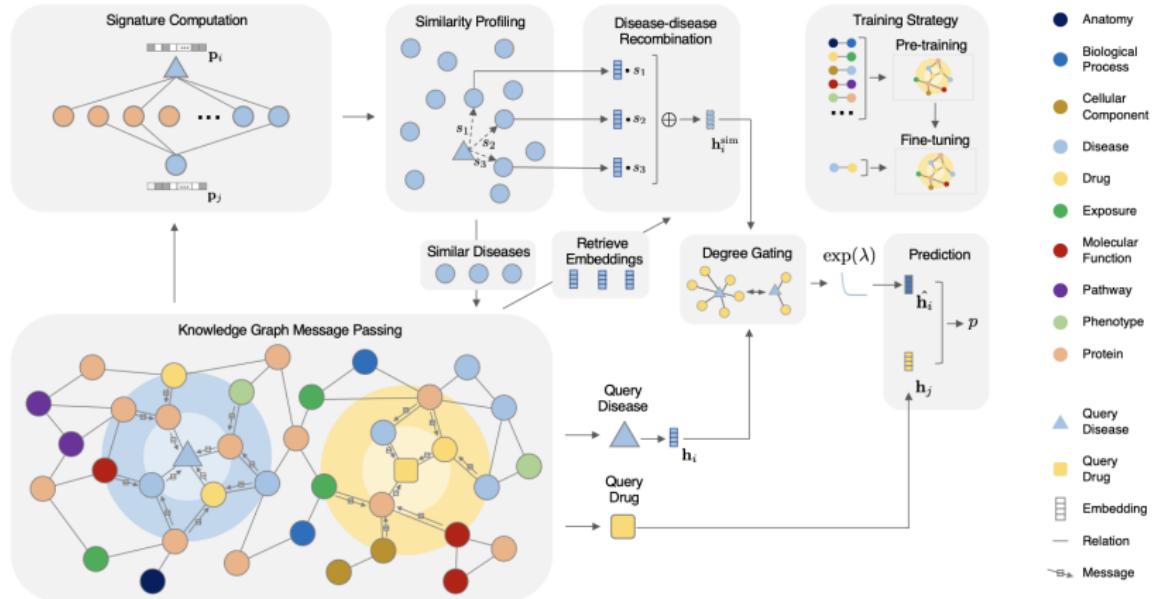


Figure: Fig. 1b from Huang (2023) [9]

Downloadable Slides

# Architecture

- ① A heterogeneous graph neural network-based encoder to obtain biologically meaningful network representation for each biomedical entity
- ② A disease similarity-based metric learning decoder to leverage auxiliary information to enrich the representation of diseases that lack molecular characterization
- ③ An all-relation stochastic pre-training followed by a drug-disease centric full-graph fine-tuning strategy
- ④ A graph explainability module to retain a sparse set of edges that are crucial for prediction as a post-training step

# Architecture

- ①  $r$ -MPNN
- ② Disease similarly quantification (and gatekeeping)
- ③ Pre-train on KG mini-batches; Fine-tune on drug-disease pairs
- ④ Interpretability

# Heterogenous graph neural network encoder

Here, let  $\mathcal{N}_r$  be the neighborhood only for edges of relation type  $r$

*Message* :  $m_{r,i} = \Theta_{r,M} h_i$

*Aggregate* :  $\widetilde{m}_{r,i} = |\mathcal{N}_{r(i)}|^{(-1)} \sum_{j \in \mathcal{N}_{r(i)}} m_{r,j}$

*Update* :  $h_i \leftarrow h_i + \sum_{r \in \mathcal{T}_R} \widetilde{m}_{r,i}$

$$p_{i,j,r} = \text{expit}\left(\sum h_i \cdot w_r \cdot h_j\right)$$

$$\mathcal{L} = \sum_{(i,r,j)} y_{i,r,j} * \log(p_{i,r,j}) + (1 - y_{i,r,j}) * \log(1 - p_{i,r,j})$$

# Disease Similarity

- Recall TxGNN's goal is zero-shot for poorly-understood diseases (i.e. likely sparsely connected)
- Rather than update these embeddings, identify most similar and well-understood diseases to utilize KG information
  - Protein structure similarity
  - Similar phenotype or exposure
- Supply a "gating mechanism" to induce similarity updating of poorly-understood diseases while discouraging updating well-connected/known diseases

# Pre-training/Fine-tuning

- Desire for PT/FT is two-fold: computation and scope
- Pre-train on random mini-batches
- Use pre-trained encoder weights (with exception of  $w_r$ ) for fine-tuning on drug-disease pairs

# Architecture (Revisited)

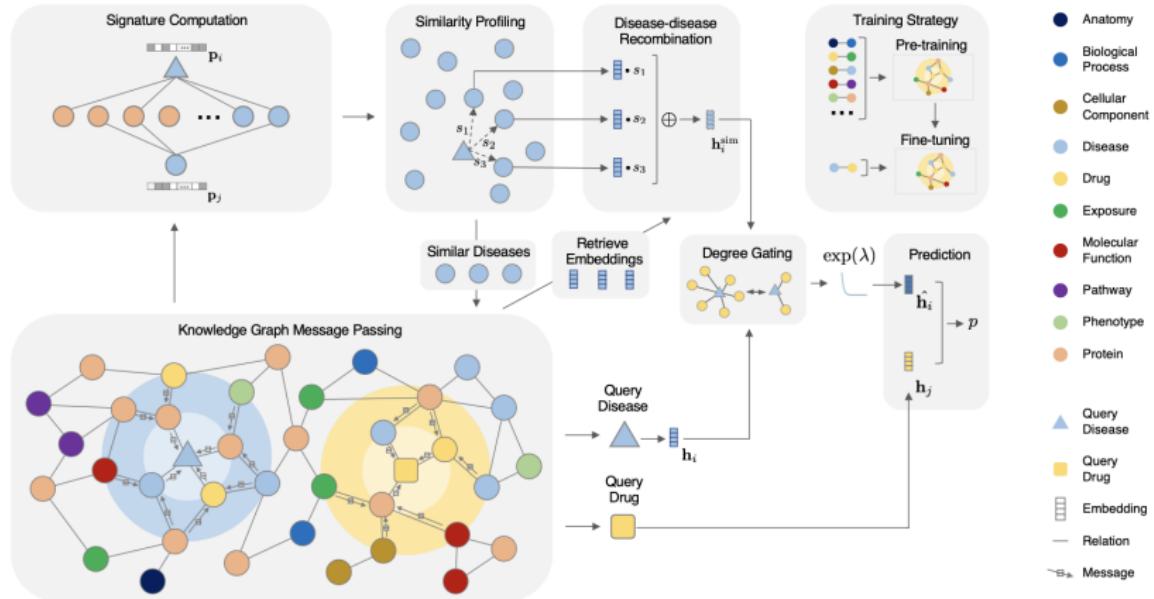


Figure: Fig. 1b from Huang (2023) [9]

# Explainability

- Account for lack of interpretability of model parameters by identifying sparse edge sets for relevant predictions

- Calculate indicator for edge inclusion

$$z^{(\kappa)} = \mathbf{1}_{[\mathbb{R} > 0.5]} \left( \text{sigmoid} \left( W_{g,r}^{(\kappa)} (m_{r,i}^{(\kappa)} || m_{r,j}^{(\kappa)}) \right) \right)$$

- Update message  $\hat{m}_{ir}^{(\kappa)} = z_{ijr}^{(\kappa)} \cdot m_{i,r}^{(\kappa)} + (1 - z_{i,j,r}^{(\kappa)}) \cdot \mathbf{b}_r^{(\kappa)}$  for learnable baseline vector  $\mathbf{b}_r$

- Train with loss function:

$$\max_{\lambda} \min_{\mathbf{W}_g} \sum_{k=1}^L \sum_{(i,j,r) \in D_+ \cup D_-} \mathbb{1}_{[R \neq 0]} z_{i,j,r}^{(k)} + \lambda (\|\hat{p}_{i,j,r} - p_{i,j,r}\|^2 - \beta)$$

# TxGNN

- Leverage heterogeneous KG data to predict therapies for poorly understood (read: sparsely KG connected) diseases
- Using large, heterogeneous KG data combined with understandings of disease similarity elucidate better understandings of rare diseases or expand use of orphan drugs
- Requires a non-trivial data harmonization effort as well as a PT/FT method for computational feasibility
- Authors offer an explainability analysis by identifying sparse edge sets with "faithful predictions"

# Conclusion

# References I

- Some diagrams generated in conjunction with ChatGPT 3.5
- [1] Nasim Abdollahi et al. *NodeCoder: a graph-based machine learning platform to predict active sites of modeled protein structures.* arXiv:2302.03590 [q-bio]. Feb. 2023.
- [2] Payal Chandak, Kexin Huang, and Marinka Zitnik. “Building a knowledge graph to enable precision medicine”. en. In: *Scientific Data* 10.1 (Feb. 2023). Number: 1 Publisher: Nature Publishing Group, p. 67.
- [3] David Duvenaud et al. *Convolutional Networks on Graphs for Learning Molecular Fingerprints.* arXiv:1509.09292 [cs, stat]. Nov. 2015.

## References II

- [4] Yasha Ektefaie et al. "Multimodal learning with graphs". en. In: *Nature Machine Intelligence* 5.4 (Apr. 2023). Number: 4 Publisher: Nature Publishing Group, pp. 340–350.
- [5] Oleksandr Ferludin et al. *TF-GNN: Graph Neural Networks in TensorFlow*. arXiv:2207.03522 [physics, stat]. July 2023.
- [6] P. Gainza et al. "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning". en. In: *Nature Methods* 17.2 (Feb. 2020). Number: 2 Publisher: Nature Publishing Group, pp. 184–192.
- [7] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. arXiv:1704.01212 [cs]. June 2017.

## References III

- [8] M. Gori, G. Monfardini, and F. Scarselli. "A new model for learning in graph domains". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*. Vol. 2. ISSN: 2161-4407. July 2005, 729–734 vol. 2.
- [9] Kexin Huang et al. *Zero-shot drug repurposing with geometric deep learning and clinician centered design*. en. Pages: 2023.03.19.23287458. Sept. 2023.
- [10] Clemens Isert, Kenneth Atz, and Gisbert Schneider. "Structure-based drug design with geometric deep learning". en. In: *Current Opinion in Structural Biology* 79 (Apr. 2023), p. 102548.
- [11] Adrián Javaloy, Maryam Meghdadi, and Isabel Valera. *Mitigating Modality Collapse in Multimodal VAEs via Impartial Optimization*. en. arXiv:2206.04496 [cs]. June 2022.

## References IV

- [12] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv:1609.02907 [cs, stat]. Feb. 2017.
- [13] Michelle M. Li, Kexin Huang, and Marinka Zitnik. “Graph representation learning in biomedicine and healthcare”. eng. In: *Nature Biomedical Engineering* 6.12 (Dec. 2022), pp. 1353–1369.
- [14] Matthew B. A. McDermott et al. “Structure-inducing pre-training”. en. In: *Nature Machine Intelligence* 5.6 (June 2023). Number: 6 Publisher: Nature Publishing Group, pp. 612–621.
- [15] Seokhyun Moon et al. “PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions”. en. In: *Chemical Science* 13.13 (2022). Publisher: Royal Society of Chemistry, pp. 3661–3673.

## References V

- [16] Benjamin Sanchez-Lengeling et al. "A Gentle Introduction to Graph Neural Networks". en. In: *Distill* 6.9 (Sept. 2021), e33.
- [17] F. Scarselli et al. "The Graph Neural Network Model". en. In: *IEEE Transactions on Neural Networks* 20.1 (Jan. 2009), pp. 61–80.
- [18] Jérôme Tubiana, Dina Schneidman-Duhovny, and Haim J. Wolfson. "ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction". en. In: *Nature Methods* 19.6 (June 2022). Number: 6 Publisher: Nature Publishing Group, pp. 730–739.
- [19] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* arXiv:1810.00826 [cs, stat]. Feb. 2019.
- [20] Liang Yao, Chengsheng Mao, and Yuan Luo. *Graph Convolutional Networks for Text Classification*. arXiv:1809.05679 [cs]. Nov. 2018.

Downloadable Slides

## References VI

- [21] Zaixi Zhang et al. *A Systematic Survey in Geometric Deep Learning for Structure-based Drug Design*. arXiv:2306.11768 [cs, q-bio]. Oct. 2023.
- [22] Shuai Zheng et al. “Multi-Modal Graph Learning for Disease Prediction”. In: *IEEE Transactions on Medical Imaging* 41.9 (Sept. 2022). Conference Name: IEEE Transactions on Medical Imaging, pp. 2207–2216.

# Appendix Slides

[Downloadable Slides](#)

# Structure-Induced Pre-Training (1/2)

- Consider a pre-training (PT)/fine-tuning (FT) problem
- We pre-train  $f_{\theta} : \mathcal{X} \rightarrow \mathcal{Z}$  on some PT data  $\mathbf{X}_{PT} \in \mathcal{X}$  and loss function  $\mathcal{L}_{PT}$
- Use the encoded  $f(\mathbf{X}_{PT}) \in \mathcal{Z}$  on some downstream, FT tasks
  - Protein folding category, stability, amino acid properties
  - Text sentiment
- Applied in language models, protein behavior prediction

## Structure-Induced Pre-Training (2/2)

- In addition to  $\mathbf{X}_{PT}$ , we also have underlying structure  $\mathcal{G}_{PT}$
- Our PT encoder  $f_\theta$  must only take  $\mathbf{X}_{PT}$  as input to allow transferability to FT tasks
- Decompose loss function as PT objective  $\mathcal{L}_{Obj}$  and structure-inducing  $\mathcal{L}_{\mathcal{G}}$ :

$$\mathcal{L}_{PT} = (1 - \lambda)\mathcal{L}_{Obj} + \lambda\mathcal{L}_{\mathcal{G}}$$

# Implementations

- PyTorch Geometric with directed extension [PyTorch Geometric](#)  
[Signed Directed](#)
- TensorFlow GNN [5]
- GraphNeuralNetworks.jl
- Spektral, Keras-based Python package
- Limited but some implementation in R
  - scapGNN, package GNN implementation but specific/narrow for single-cell -omics data

# Abbreviated History

- Graph Neural Network first(?) coined in Gori (2005) [8] and subsequently in Scarselli (2009) *The Graph Neural Network Model* [17]
- Graph Convolutional Network by Kipf (2017) [12] but with similar convolutional message-passing algorithms (within GNN's) proposed in at least 2015 [3]
- Message passing GNN proposed in Gilmer (2017), applications in molecular chemistry [7]

# Additional Applications, Interesting Papers I

- Applications in travel time prediction, 2021  
(<https://arxiv.org/pdf/2108.11482.pdf>)
- Someone has compiled graph-/GNN-relevant talks for NeurIPS 2023 at [https://github.com/XiaoxinHe/neurips2023\\_learning\\_on\\_graphs](https://github.com/XiaoxinHe/neurips2023_learning_on_graphs)

# NodeCoder Embeddings

**Table S2** Node features.

feature name	description	count
<b>Amino Acid</b>	Twenty binary features to encode the node's corresponding amino acid (one-hot encoding)	20
<b>iPlus</b>	Amino acid immediately prior to this node in the protein's primary sequence	1
<b>iMinus</b>	Amino acid immediately following this node in the protein's primary sequence	1
<b>DSSP</b>	Seven features encoding DSSP secondary structure assignment (one-hot encoding) and eight features that encode DSSP H-bonding interactions	15
<b>dihedral angles</b>	dihedral angles: $\phi, \psi, \tau, \theta$	4
<b>BBSASA</b>	Back Bone Solvent Accessibility	1
<b>SCSASA</b>	Side Chain Solvent Accessibility	1
<b>pLDDT</b>	AlphaFold per-residue confidence metric [5]	1
<b>centric distance</b>	euclidean distance of amino acid residue from the center of protein	1
<b>centric cosine distance</b>	cosine distance of amino acid residue from the center of protein	1
<b>centrality*</b>	node degree capturing the node importance in the graph	1

\*This feature is obtained from graph representation of protein.

Figure: From Table S2 in Abdollahi (2023) [1]

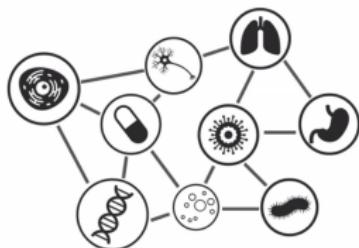
Downloadable Slides

# KG AI Models

Full figure from McDermott et al. [14], cropped and presented in Introduction:

## Knowledge graph AI models

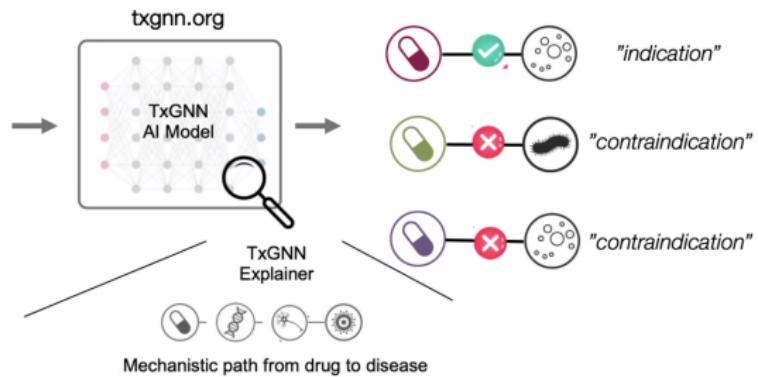
Multimodal knowledge graph  
of 17,080 disease phenotypes



Semi-automatic KG rebuild when new datasets become available

Building a knowledge graph to enable precision medicine, Scientific Data 2023

Process therapeutic tasks and predict candidate indications and contraindications



Structure-inducing pre-training, Nature Machine Intelligence 2023; Multimodal learning with graphs, Nature Machine Intelligence 2023;  
Zero-shot prediction of therapeutic use with geometric deep learning and clinician centered design, medRxiv, 2023

Downloadable Slides



# Protein Data Bank File

```

HEADER EXTRACELLULAR MATRIX           22-JAN-98   1A3I
TITLE X-RAY CRYSTALLOGRAPHIC DETERMINATION OF A COLLAGEN-LIKE
TITLE 2 PEPTIDE WITH THE REPEATING SEQUENCE (PRO-PRO-GLY)
...
EXPDTA X-RAY DIFFRACTION
AUTHOR R.Z.KRAMER,L.VITAGLIANO,J.BELLA,R.BERISIO,L.MAZZARELLA,
AUTHOR 2 B.BRODSKY,A.ZAGARI,H.M.BERMAN
...
REMARK 350 BIOMOLECULE: 1
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B, C
REMARK 350 BIOMT1   1  1.000000  0.000000  0.000000      0.00000
REMARK 350 BIOMT2   1  0.000000  1.000000  0.000000      0.00000
...
SEQRES 1 A    9  PRO PRO GLY PRO PRO GLY PRO PRO GLY
SEQRES 1 B    6  PRO PRO GLY PRO PRO GLY
SEQRES 1 C    6  PRO PRO GLY PRO PRO GLY
...
ATOM    1  N    PRO A    1       8.316  21.206  21.530  1.00 17.44      N
ATOM    2  CA   PRO A    1       7.608  20.729  20.336  1.00 17.44      C
ATOM    3  C    PRO A    1       8.487  20.707  19.092  1.00 17.44      C
ATOM    4  O    PRO A    1       9.466  21.457  19.005  1.00 17.44      O
ATOM    5  CB   PRO A    1       6.460  21.723  20.211  1.00 22.26      C
...
HETATM 130  C    ACY   401     3.682  22.541  11.236  1.00 21.19      C
HETATM 131  O    ACY   401     2.807  23.097  10.553  1.00 21.19      O
HETATM 132  OXT  ACY   401     4.306  23.101  12.291  1.00 21.19      O
...

```