Notes/Questions (To Be Deleted) I

- Touch on geometric DL in general? Or just focus on GNN's and not worry about placing within the DL landscape?
- Touch on AlphaFold? GNN's used I believe but unconfirmed, and architecture is complicated/seems infeasible (at least for my time right now)
- Still unsure how to tie in PrimeKG paper
- Node-level seems intuitive, my pseudocode already outlines iterative updating of node features and we mention reg/class/multiclass
- Methods for edge selection/identification? Some instances we only want to preserve structure, but sometimes we want to predict/fit edges, how does that fit into message-passing/GCN methods or other extensions?
- Example of graph-level classification?

Downloadable Slides

Notes/Questions (To Be Deleted) II

- Diffpool article as extension of general GNN, hierarchical pooling over global readout functions?
  https://papers.nips.cc/paper_files/paper/2018/file/e77dbaf6759253c7c6d0efc5690369c7-Paper.pdf

Downloadable Slides

Graphical Neural Networks

Dominic DiSanto

Department of Biostatistics, Harvard University

November 28, 2023

Downloadable Slides

Outline

1. Set-Up and Motivation 5ish minutes
2. General Construction 10-20ish minutes
3. Applications and Extensions 20-25ish minutes
   - Knowledge-Graph Data
   - Multimodal Biomedical Data

Downloadable Slides

Goals

- Provide a useful overview of Graphical Neural Networks (GNN)
  - Motivation for necessity of GNN's
  - Provide a general framework of fitting
- Describe applications and extensions of the general GNN model
  - Multimodal Physiological/Biomedical Data
  - Integration of Knowledge-Graph and EHR Data

Downloadable Slides

KG Application

Image here

Downloadable Slides

Multi-modal Biomedical Data

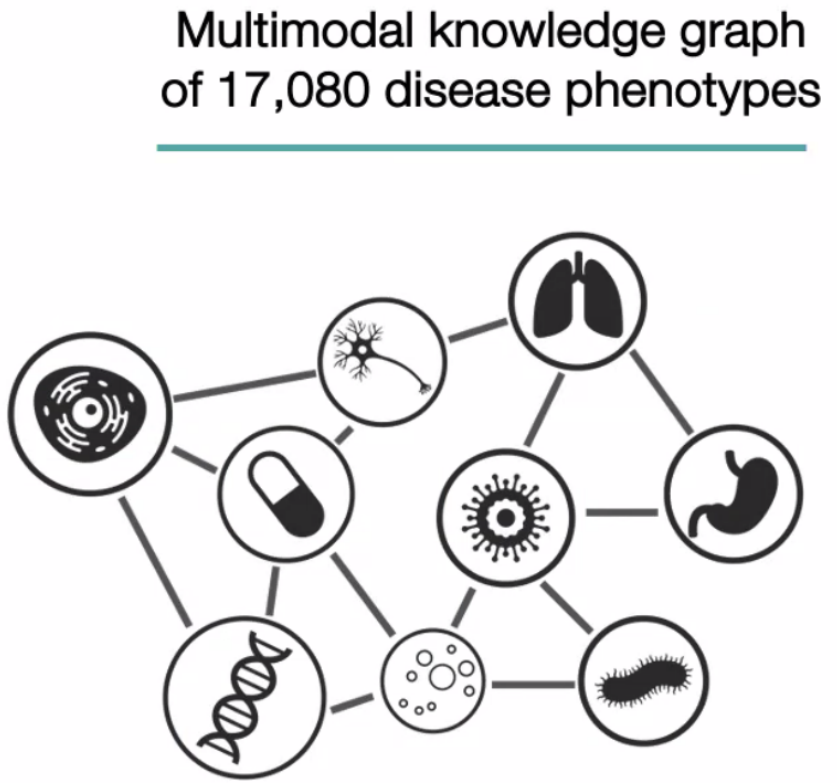
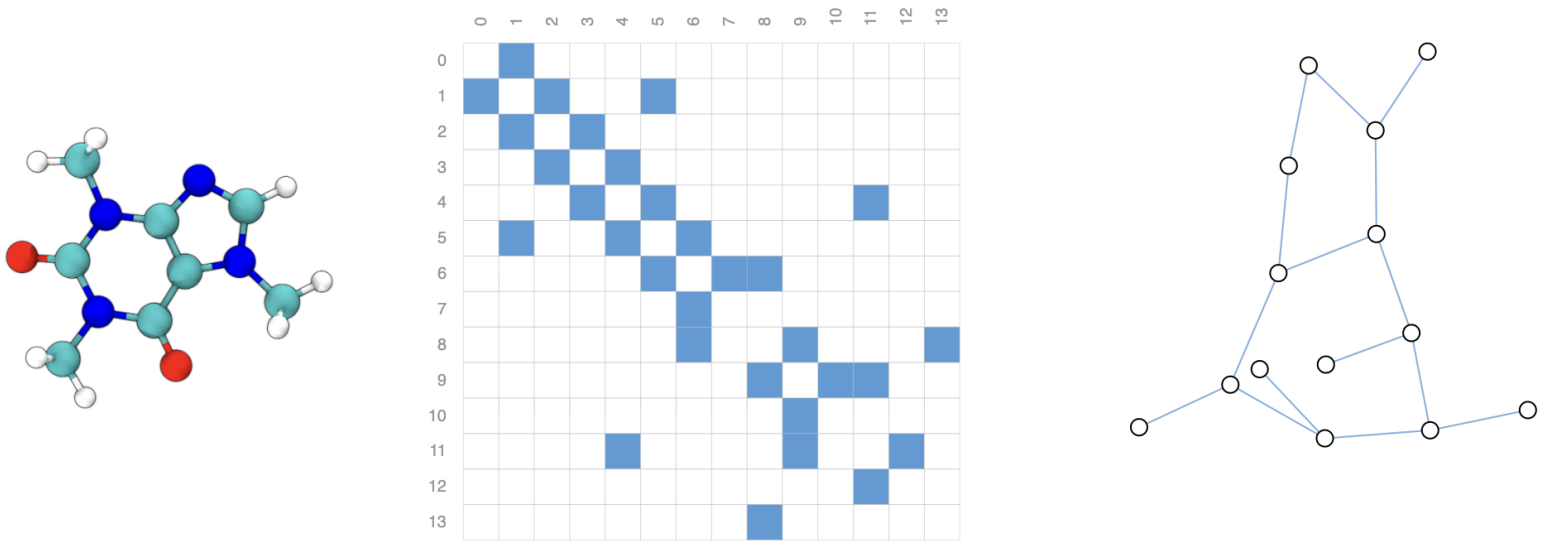


Image courtesy of partial figure from McDermott et al. *Structure-inducing pre-training* [7]

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/ [8]

Downloadable Slides

Protein Folding Maybe drop? 4 examples probably unnecessary

AlphaFold image/example here if I can find a nice one

Downloadable Slides

Motivation

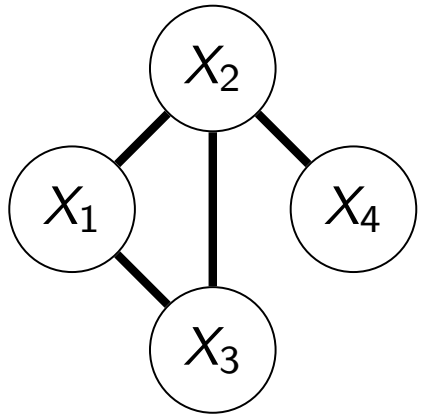
- Want to utilize the input structure of the graph
  - Respect/Maintain
  - Update/Estimate
- "Flattening" graphical data for DNN, CNN, etc. omits useful topology from our data
- Early methods attempting to retain topological info included recursive neural networks and random walk models, which GNN methodology extended [9]

Downloadable Slides

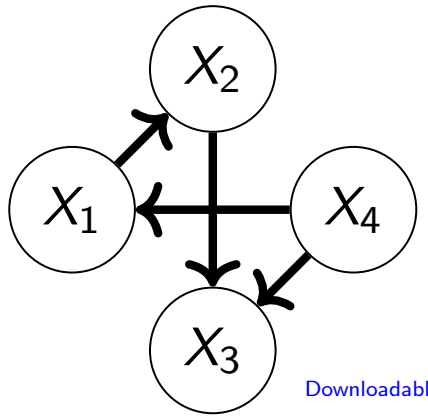
Notation/Set-Up

- 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 $N = N$
- 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 $\mathbf{L} = \mathbf{D} - \mathbf{A}$
  - $\mathbf{D} = \text{diag}(\mathbf{A1_N})$
- Can use an eigenvalue-normalized Laplacian $\widetilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$
- Define graph convolution here?
- Note on necessity of permutation invariant functions (permutations of adjacency matrix represent same graph but consistent behavior of NN/GNN is not assured by permutd A matrices)
- Adjacency matrix can be prohibitively large but likely sparse, is also permutable but DNN's are not permutation invariant, undesirable
- Can store an adjacency list

Downloadable Slides

Outline

1. Set-Up and Motivation 5ish minutes
2. General Construction 10-20ish minutes
3. Applications and Extensions 20-25ish minutes
   - Knowledge-Graph Data
   - Multimodal Biomedical Data

Downloadable Slides

What do we estimate about graph structure?

See supp note 1 on Multimodal learning with graphs

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

Downloadable Slides

What do we estimate about graph structure?

Planning (very tentatively) to note that I focus of node-wise estimation as primary interest. My understanding (with still much to read/learn) is that edge/graph-level characteristics are then essentially secondary outputs of node-characteristics. i.e. we fit the GNN, predict node labels/values/embeddings, then fit separate models to predict edge-/graph-level information

GNN learning can be
- Node-wise $\Phi(\mathcal{G}, x) : (x \in V) \to \mathbb{R}^m$
- Graph-level characteristics $\Phi(\mathcal{G}, e) : (e \in E) \to \mathbb{R}^m$
- Graph-level characteristics $\Phi(\mathcal{G})$

Downloadable Slides

General Framework[1]

Title NodeUpdate and create second slide with more clear backpropagation training (i.e. iter over NodeUpdate)

General structure:

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

[1] See [2, 4, 10]

Downloadable Slides

General Framework

Can succinctly represent the κth layer as:

$$\mathbf{h}_v^{(\kappa+1)} = \text{Update}\left(x_v^{(\kappa)}, \text{Aggregate}(h_v^{(\kappa)}, x_u^{(\kappa)}, e_{u,v}^{(\kappa)})\right)$$

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

Trained end-to-end via backpropagation

Downloadable Slides

Aggregate & Update

- **Aggregate**$(\cdot)$ produces a represntation of information from a node's neighborhood
  - Also **Message**$(\cdot)$ in the context of Message Passing Neural Networks (MPNN's) [4]
- Differentiable, *permutation invariant* functions
- 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**$(\cdot)$ our current state using this aggregated neighborhood-level information

Downloadable Slides

Transform/Readout

- **Transform**$(\cdot)$ translates our learned node representations to some desired outcome
  - Regression
  - Binary/Multi-class classification
  - MLP/DNN's
- **Readout**$(\cdot)$ 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.)

Downloadable Slides

Message Passing

- Our **Aggregate**$(\cdot)$ step can also be thought

Downloadable Slides

Graph Convolutional Network

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

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

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

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

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

Downloadable Slides

Graph Convolutional Network

Proposed in 2017 by Thomas Kipf, Max Welling [6]

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

- Can replace ReLu with any activation function (e.g. sigmoid, GELU, etc.)
- Learned weight/parameter matrix $\mathbf{\Theta}$

Review paper, comment on applications briefly (KG setting), also motivate through graph convolutions?

Downloadable Slides

Graph Convolutional Network

Proposed in 2017 by Thomas Kipf, Max Welling [6]

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

- Can replace ReLu with any activation function (e.g. sigmoid, GELU, etc.)
- Learned weight/parameter matrix $\mathbf{\Theta}$

Review paper, comment on applications briefly (KG setting), also motivate through graph convolutions?

Downloadable Slides

Outline

1. Set-Up and Motivation 5ish minutes
2. General Construction 10-20ish minutes
3. Applications and Extensions 20-25ish minutes
   - Knowledge-Graph Data
   - Multimodal Biomedical Data

Downloadable Slides

Motivation

Downloadable Slides

Conclusion

Downloadable Slides

References I

- Some diagrams generated in conjunction with ChatGPT 3.5

[1] David Duvenaud et al. *Convolutional Networks on Graphs for Learning Molecular Fingerprints*. arXiv:1509.09292 [cs, stat]. Nov. 2015.

[2] 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.

[3] Oleksandr Ferludin et al. *TF-GNN: Graph Neural Networks in TensorFlow*. arXiv:2207.03522 [physics, stat]. July 2023.

[4] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. arXiv:1704.01212 [cs]. June 2017.

Downloadable Slides

References II

[5] 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.

[6] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv:1609.02907 [cs, stat]. Feb. 2017.

[7] 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.

[8] Benjamin Sanchez-Lengeling et al. "A Gentle Introduction to Graph Neural Networks". en. In: *Distill* 6.9 (Sept. 2021), e33.

Downloadable Slides

References III

[9] F. Scarselli et al. "The Graph Neural Network Model". en. In: *IEEE Transactions on Neural Networks* 20.1 (Jan. 2009), pp. 61–80.

[10] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* arXiv:1810.00826 [cs, stat]. Feb. 2019.

[11] Zhitao Ying et al. "Hierarchical Graph Representation Learning with Differentiable Pooling". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

Downloadable Slides

Appendix Slides

Downloadable Slides

Implementations

- PyTorch Geometric with directed extension PyTorch Geometric Signed Directed
- TensorFlow GNN [3]
- GraphNeuralNetworks.jl
- Spektral, Keras-based Pythhon package
- Limited but some implementation in R
  - scapGNN, package GNN implementation but specific/narrow for single-cell -omics data

Downloadable Slides

Abbreviated History

- Graph Neural Network first(?) coined in Gori (2005) [5] and subqeuently in Scarselli (2009) *The Graph Neural Network Model* [9]
- Graph Convolutional Network by Kipf (2017) [6] but with similar convolutional message-passing algorithms (within GNN's) proposed in at least 2015 [1]
- Message passing GNN proposed in Gilmer (2017), applications in molecular chemistry [4]

Downloadable Slides

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

DiffPool

Figure 1 from Ying (2018) *Hierarchical Graph Representation Learning with Differentiable Pooling* [11]

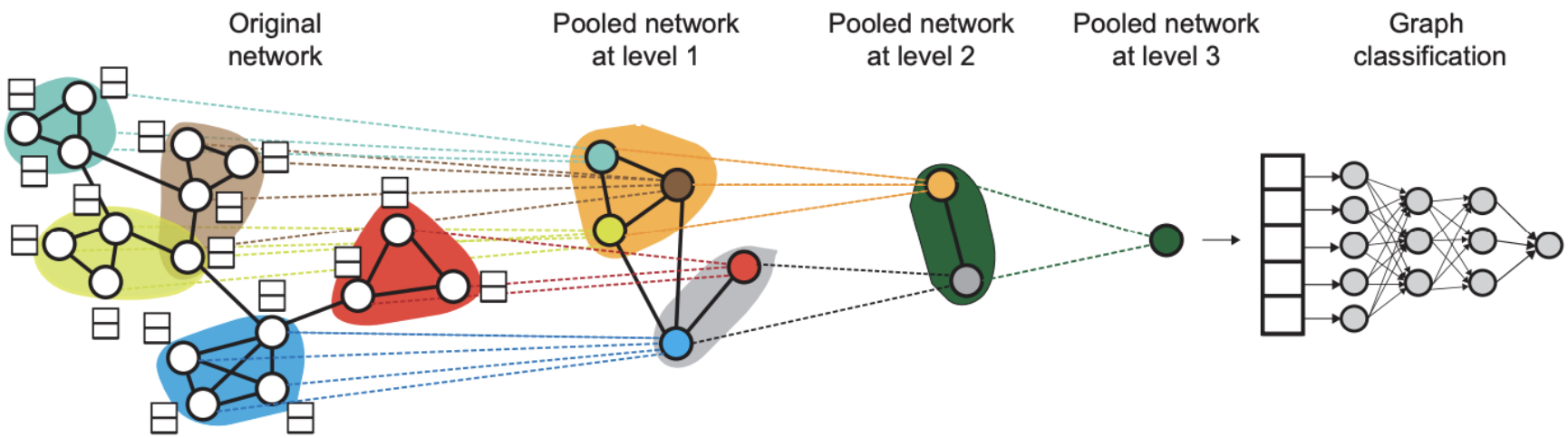
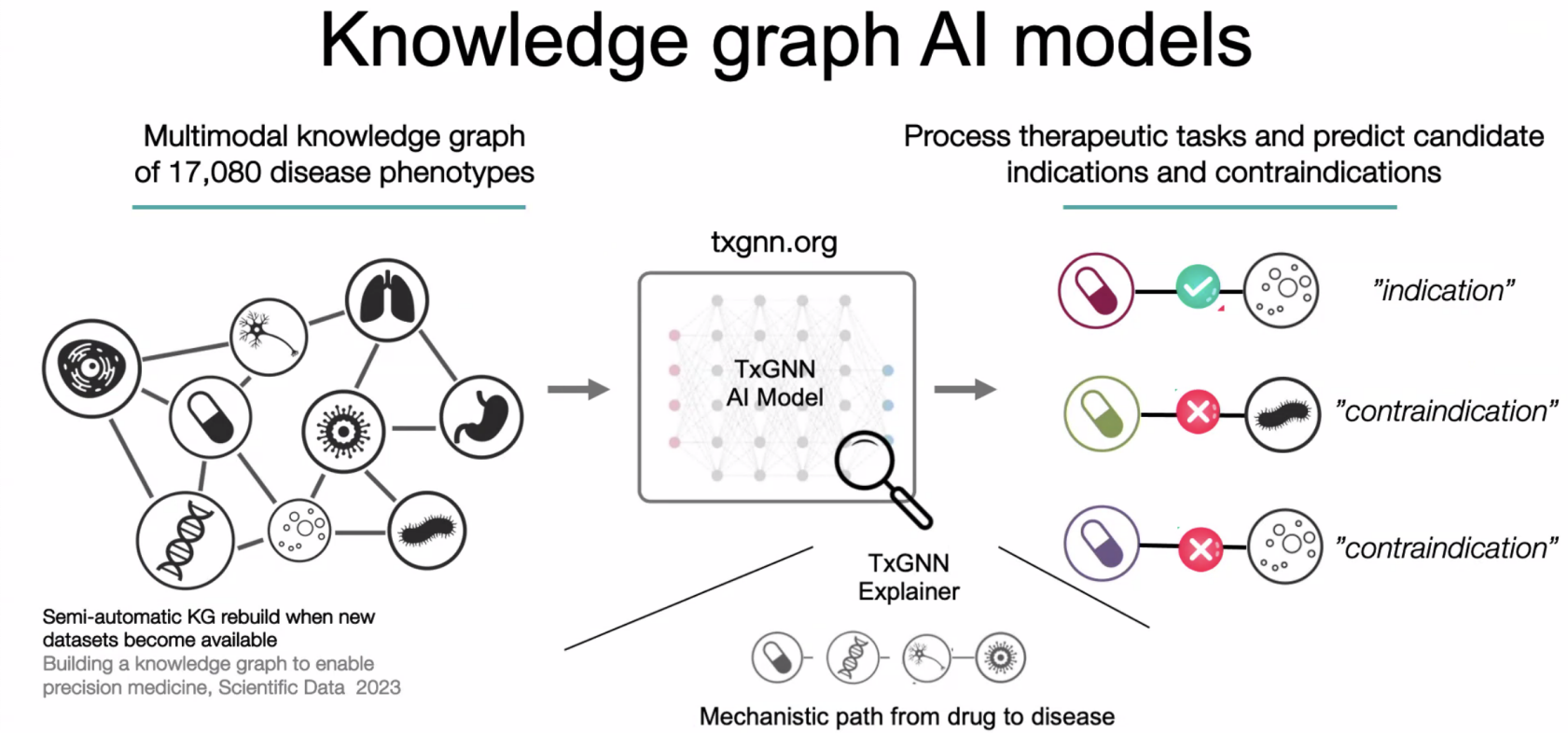


Figure 1: High-level illustration of our proposed method DIFFPOOL. At each hierarchical layer, we run a GNN model to obtain embeddings of nodes. We then use these learned embeddings to cluster nodes together and run another GNN layer on this coarsened graph. This whole process is repeated for L layers and we use the final output representation to classify the graph.

Downloadable Slides

KG AI Models

Full figure from McDermott et al. [7], cropped and presented in Introduction:



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