

Conclusion

Jiaxuan You

Assistant Professor at UIUC CDS



CS598: Deep Learning with Graphs, 2024 Fall

<https://ulab-uiuc.github.io/CS598/>

Review Our Course Objective

Takeaways from this course:

- **Knowledge** about graph deep learning
 - Core knowledge: Insights, Coding, Math
 - Latest knowledge: Recent research papers
- **Training** for AI research
 - Experience the **full lifecycle as an AI researcher**
 - Read, Ideate, Discuss, Code, Write, Review, Present

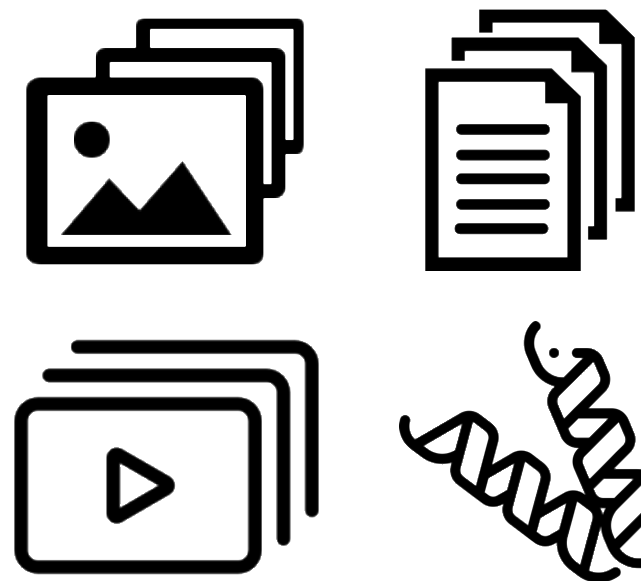
Conclusion

Knowledge about graph deep learning



Interconnected world

Gap
↔



Modern ML

Goal of Graph Deep Learning

Enable DL research for the interconnected data

Chemical structure of N-(1,1-diphenylpropyl)propanamide. The structure shows a central carbon atom bonded to two phenyl rings and a propyl chain. The propyl chain is terminated by an amide group (-CONH₂).

The diagram illustrates a complex network of drug-target interactions. Drugs are represented by yellow squares and targets by purple circles. The network is highly interconnected with numerous edges. The diagram is framed by a red bracket on the left side.

Drugs (Yellow Squares): Terazosin, Alfuzosin, Midodrine, Tolterodine, Reserpine, Nicardipine, Imipramine, Fluoxetine, Sildenafil, Tadalafil, Prazosin, Thioridazine, Risperidone, Chlorpromazine, Doxepin, Labetalol, Tetrabenazine, Clonidine, Bisoprolol, Metoprolol, Timolol, Pindolol, Isoprenaline, Carteolol, Imatinib, Propranolol, Almotriptan, Rizatriptan, Betaxolol, Entacapone, Ondansetron, Citalopram, Sotalol, Amoxapine, Zolmitriptan, Methyldopa, Latanoprost, Apomorphine, Ergotamine, Pramipexole, Triazodone, Desflurane, Phenobarbital, Donepezil, Galantamine, Enflurane, Estazolam, Triazolam, Midazolam, Lorazepam, Isoflurane, Bromazepam, Clobazam, Nitrazepam, Alprazolam, Oxazepam, Flurazepam, Pentobarbital, Sevoflurane, Primidone, Clonazepam, Chlorazepoxide, Gabapentin, Pregabalin, Desflurane, Phenobarbital, Donepezil, Galantamine, Enflurane, Estazolam, Triazolam, Midazolam, Lorazepam, Isoflurane, Bromazepam, Clobazam, Nitrazepam, Alprazolam, Oxazepam, Flurazepam, Pentobarbital, Sevoflurane, Primidone, Clonazepam, Chlorazepoxide, Gabapentin, Pregabalin.

Targets (Purple Circles): SLC6A1, ADRA1A, CHRM1, GNG2, SLC6A3, ADRA1D, CHRM3, SLC9A1, ADRA1B, CHRM4, CHRM5, ADRA2A, ADRA2B, HTR1A, HTR1B, HTR1D, HTR2A, HTR2B, ADRA2C, ADRA2D, ADRA2E, ADRA2F, ADRA2G, ADRA2H, ADRA2I, ADRA2J, ADRA2K, ADRA2L, ADRA2M, ADRA2N, ADRA2O, ADRA2P, ADRA2Q, ADRA2R, ADRA2S, ADRA2T, ADRA2U, ADRA2V, ADRA2W, ADRA2X, ADRA2Y, ADRA2Z, ADRA2AA, ADRA2AB, ADRA2AC, ADRA2AD, ADRA2AE, ADRA2AF, ADRA2AG, ADRA2AH, ADRA2AI, ADRA2AJ, ADRA2AK, ADRA2AL, ADRA2AM, ADRA2AN, ADRA2AO, ADRA2AP, ADRA2AQ, ADRA2AR, ADRA2AS, ADRA2AT, ADRA2AU, ADRA2AV, ADRA2AW, ADRA2AX, ADRA2AY, ADRA2AZ, ADRA2BA, ADRA2BB, ADRA2BC, ADRA2BD, ADRA2BE, ADRA2BF, ADRA2BG, ADRA2BH, ADRA2BI, ADRA2BJ, ADRA2BK, ADRA2BL, ADRA2BM, ADRA2BN, ADRA2BO, ADRA2BP, ADRA2BQ, ADRA2BR, ADRA2BS, ADRA2BT, ADRA2BU, ADRA2BV, ADRA2BW, ADRA2BX, ADRA2BY, ADRA2BZ, ADRA2CA, ADRA2CB, ADRA2CC, ADRA2CD, ADRA2CE, ADRA2CF, ADRA2CG, ADRA2CH, ADRA2CI, ADRA2CJ, ADRA2CK, ADRA2CL, ADRA2CM, ADRA2CN, ADRA2CO, ADRA2CP, ADRA2CQ, ADRA2CR, ADRA2CS, ADRA2CT, ADRA2CU, ADRA2CV, ADRA2CW, ADRA2CX, ADRA2CY, ADRA2CZ, ADRA2DA, ADRA2DB, ADRA2DC, ADRA2DD, ADRA2DE, ADRA2DF, ADRA2DG, ADRA2DH, ADRA2DI, ADRA2DJ, ADRA2DK, ADRA2DL, ADRA2DM, ADRA2DN, ADRA2DO, ADRA2DP, ADRA2DQ, ADRA2DR, ADRA2DS, ADRA2DT, ADRA2DU, ADRA2DV, ADRA2DW, ADRA2DX, ADRA2DY, ADRA2DZ, ADRA2EA, ADRA2EB, ADRA2EC, ADRA2ED, ADRA2EE, ADRA2EF, ADRA2EG, ADRA2EH, ADRA2EI, ADRA2EJ, ADRA2EK, ADRA2EL, ADRA2EM, ADRA2EN, ADRA2EO, ADRA2EP, ADRA2EQ, ADRA2ER, ADRA2ES, ADRA2ET, ADRA2EU, ADRA2EV, ADRA2EW, ADRA2EX, ADRA2EY, ADRA2EZ, ADRA2FA, ADRA2FB, ADRA2FC, ADRA2FD, ADRA2FE, ADRA2FF, ADRA2FG, ADRA2FH, ADRA2FI, ADRA2FJ, ADRA2FK, ADRA2FL, ADRA2FM, ADRA2FN, ADRA2FO, ADRA2FP, ADRA2FQ, ADRA2FR, ADRA2FS, ADRA2FT, ADRA2FU, ADRA2FV, ADRA2FW, ADRA2FX, ADRA2FY, ADRA2FZ, ADRA2GA, ADRA2GB, ADRA2GC, ADRA2GD, ADRA2GE, ADRA2GF, ADRA2GG, ADRA2GH, ADRA2GI, ADRA2GJ, ADRA2GK, ADRA2GL, ADRA2GM, ADRA2GN, ADRA2GO, ADRA2GP, ADRA2GQ, ADRA2GR, ADRA2GS, ADRA2GT, ADRA2GU, ADRA2GV, ADRA2GW, ADRA2GX, ADRA2GY, ADRA2GZ, ADRA2HA, ADRA2HB, ADRA2HC, ADRA2HD, ADRA2HE, ADRA2HF, ADRA2HG, ADRA2HH, ADRA2HI, ADRA2HJ, ADRA2HK, ADRA2HL, ADRA2HM, ADRA2HN, ADRA2HO, ADRA2HP, ADRA2HQ, ADRA2HR, ADRA2HS, ADRA2HT, ADRA2HU, ADRA2HV, ADRA2HW, ADRA2HX, ADRA2HY, ADRA2HZ, ADRA2IA, ADRA2IB, ADRA2IC, ADRA2ID, ADRA2IE, ADRA2IF, ADRA2IG, ADRA2IH, ADRA2II, ADRA2IJ, ADRA2IK, ADRA2IL, ADRA2IM, ADRA2IN, ADRA2IO, ADRA2IP, ADRA2IQ, ADRA2IR, ADRA2IS, ADRA2IT, ADRA2IU, ADRA2IV, ADRA2IW, ADRA2IX, ADRA2IY, ADRA2IZ, ADRA2JA, ADRA2JB, ADRA2JC, ADRA2JD, ADRA2JE, ADRA2JF, ADRA2JG, ADRA2JH, ADRA2JI, ADRA2JJ, ADRA2JK, ADRA2JL, ADRA2JM, ADRA2JN, ADRA2JO, ADRA2JP, ADRA2JQ, ADRA2JR, ADRA2JS, ADRA2JT, ADRA2JU, ADRA2JV, ADRA2JW, ADRA2JX, ADRA2JY, ADRA2JZ, ADRA2KA, ADRA2KB, ADRA2KC, ADRA2KD, ADRA2KE, ADRA2KF, ADRA2KG, ADRA2KH, ADRA2KI, ADRA2KJ, ADRA2KK, ADRA2KL, ADRA2KM, ADRA2KN, ADRA2KO, ADRA2KP, ADRA2KQ, ADRA2KR, ADRA2KS, ADRA2KT, ADRA2KU, ADRA2KV, ADRA2KW, ADRA2KX, ADRA2KY, ADRA2KZ, ADRA2LA, ADRA2LB, ADRA2LC, ADRA2LD, ADRA2LE, ADRA2LF, ADRA2LG, ADRA2LH, ADRA2LI, ADRA2LJ, ADRA2LK, ADRA2LL, ADRA2LM, ADRA2LN, ADRA2LO, ADRA2LP, ADRA2LQ, ADRA2LR, ADRA2LS, ADRA2LT, ADRA2LU, ADRA2LV, ADRA2LW, ADRA2LX, ADRA2LY, ADRA2LZ, ADRA2MA, ADRA2MB, ADRA2MC, ADRA2MD, ADRA2ME, ADRA2MF, ADRA2MG, ADRA2MH, ADRA2MI, ADRA2MJ, ADRA2MK, ADRA2ML, ADRA2MN, ADRA2MO, ADRA2MP, ADRA2MQ, ADRA2MR, ADRA2MS, ADRA2MT, ADRA2MU, ADRA2MV, ADRA2MW, ADRA2MX, ADRA2MY, ADRA2MZ, ADRA2NA, ADRA2NB, ADRA2NC, ADRA2ND, ADRA2NE, ADRA2NF, ADRA2NG, ADRA2NH, ADRA2NI, ADRA2NJ, ADRA2NK, ADRA2NL, ADRA2NM, ADRA2NO, ADRA2NP, ADRA2NQ, ADRA2NR, ADRA2NS, ADRA2NT, ADRA2NU, ADRA2NV, ADRA2NW, ADRA2NX, ADRA2NY, ADRA2NZ, ADRA2OA, ADRA2OB, ADRA2OC, ADRA2OD, ADRA2OE, ADRA2OF, ADRA2OG, ADRA2OH, ADRA2OI, ADRA2OJ, ADRA2OK, ADRA2OL, ADRA2OM, ADRA2ON, ADRA2OO, ADRA2OP, ADRA2OQ, ADRA2OR, ADRA2OS, ADRA2OT, ADRA2OU, ADRA2OV, ADRA2OW, ADRA2OX, ADRA2OY, ADRA2OZ, ADRA2PA, ADRA2PB, ADRA2PC, ADRA2PD, ADRA2PE, ADRA2PF, ADRA2PG, ADRA2PH, ADRA2PI, ADRA2PJ, ADRA2PK, ADRA2PL, ADRA2PM, ADRA2PN, ADRA2PO, ADRA2PP, ADRA2PQ, ADRA2PR, ADRA2PS, ADRA2PT, ADRA2PU, ADRA2PV, ADRA2PW, ADRA2PX, ADRA2PY, ADRA2PZ, ADRA2QA, ADRA2QB, ADRA2QC, ADRA2QD, ADRA2QE, ADRA2QF, ADRA2QG, ADRA2QH, ADRA2QI, ADRA2QJ, ADRA2QK, ADRA2QL, ADRA2QM, ADRA2QN, ADRA2QO, ADRA2QP, ADRA2QQ, ADRA2QR, ADRA2QS, ADRA2QT, ADRA2QU, ADRA2QV, ADRA2QW, ADRA2QX, ADRA2QY, ADRA2QZ, ADRA2RA, ADRA2RB, ADRA2RC, ADRA2RD, ADRA2RE, ADRA2RF, ADRA2RG, ADRA2RH, ADRA2RI, ADRA2RJ, ADRA2RK, ADRA2RL, ADRA2RM, ADRA2RN, ADRA2RO, ADRA2RP, ADRA2RQ, ADRA2RR, ADRA2RS, ADRA2RT, ADRA2RU, ADRA2RV, ADRA2RW, ADRA2RX, ADRA2RY, ADRA2RZ, ADRA2SA, ADRA2SB, ADRA2SC, ADRA2SD, ADRA2SE, ADRA2SF, ADRA2SG, ADRA2SH, ADRA2SI, ADRA2SJ, ADRA2SK, ADRA2SL, ADRA2SM, ADRA2SN, ADRA2SO, ADRA2SP, ADRA2SQ, ADRA2SR, ADRA2SS, ADRA2ST, ADRA2SU, ADRA2SV, ADRA2SW, ADRA2SX, ADRA2SY, ADRA2SZ, ADRA2TA, ADRA2TB, ADRA2TC, ADRA2TD, ADRA2TE, ADRA2TF, ADRA2TG, ADRA2TH, ADRA2TI, ADRA2TJ, ADRA2TK, ADRA2TL, ADRA2TM, ADRA2TN, ADRA2TO, ADRA2TP, ADRA2TQ, ADRA2TR, ADRA2TS, ADRA2TT, ADRA2TU, ADRA2TV, ADRA2TW, ADRA2TX, ADRA2TY, ADRA2TZ, ADRA2UA, ADRA2UB, ADRA2UC, ADRA2UD, ADRA2UE, ADRA2UF, ADRA2UG, ADRA2UH, ADRA2UI, ADRA2UJ, ADRA2UK, ADRA2UL, ADRA2UM, ADRA2UN, ADRA2UO, ADRA2UP, ADRA2UQ, ADRA2UR, ADRA2US, ADRA2UT, ADRA2UU, ADRA2UV, ADRA2UW, ADRA2UX, ADRA2UY, ADRA2UZ, ADRA2VA, ADRA2VB, ADRA2VC, ADRA2VD, ADRA2VE, ADRA2VF, ADRA2VG, ADRA2VH, ADRA2VI, ADRA2VJ, ADRA2VK, ADRA2VL, ADRA2VM, ADRA2VN, ADRA2VO, ADRA2VP, ADRA2VQ, ADRA2VR, ADRA2VS, ADRA2VT, ADRA2VU, ADRA2VV, ADRA2VW, ADRA2VX, ADRA2VY, ADRA2VZ, ADRA2WA, ADRA2WB, ADRA2WC, ADRA2WD, ADRA2WE, ADRA2WF, ADRA2WG, ADRA2WH, ADRA2WI, ADRA2WJ, ADRA2WK, ADRA2WL, ADRA2WM, ADRA2WN, ADRA2WO, ADRA2WP, ADRA2WQ, ADRA2WR, ADRA2WS, ADRA2WT, ADRA2WU, ADRA2WV, ADRA2WW, ADRA2WX, ADRA2WY, ADRA2WZ, ADRA2XA, ADRA2XB, ADRA2XC, ADRA2XD, ADRA2XE, ADRA2XF, ADRA2XG, ADRA2XH, ADRA2XI, ADRA2XJ, ADRA2XK, ADRA2XL, ADRA2XM, ADRA2XN, ADRA2XO, ADRA2XP, ADRA2XQ, ADRA2XR, ADRA2XS, ADRA2XT, ADRA2XU, ADRA2XV, ADRA2XW, ADRA2XX, ADRA2XY, ADRA2XZ, ADRA2YA, ADRA2YB, ADRA2YC, ADRA2YD, ADRA2YE, ADRA2YF, ADRA2YG, ADRA2YH, ADRA2YI, ADRA2YJ, ADRA2YK, ADRA2YL, ADRA2YM, ADRA2YN, ADRA2YO, ADRA2YP, ADRA2YQ, ADRA2YR, ADRA2YS, ADRA2YT, ADRA2YU, ADRA2YV, ADRA2YW, ADRA2YX, ADRA2YY, ADRA2YZ, ADRA2ZA, ADRA2ZB, ADRA2ZC, ADRA

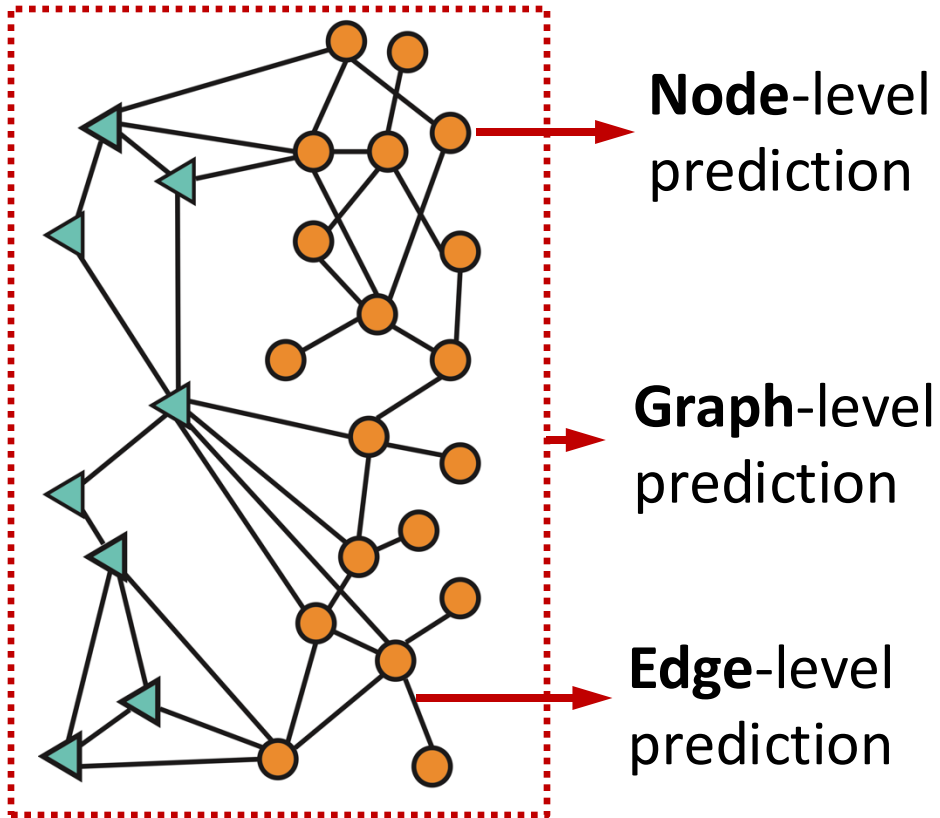
A network diagram consisting of approximately 20 circular nodes, each containing a stylized human figure. The nodes are interconnected by a web of dashed lines in various colors (blue, green, yellow, red, purple). The connections are dense in the center and become sparser towards the periphery. On the right side of the diagram, there is a large red bracket that spans the vertical extent of the network, pointing towards the text 'Network' in the adjacent column.

Economic network

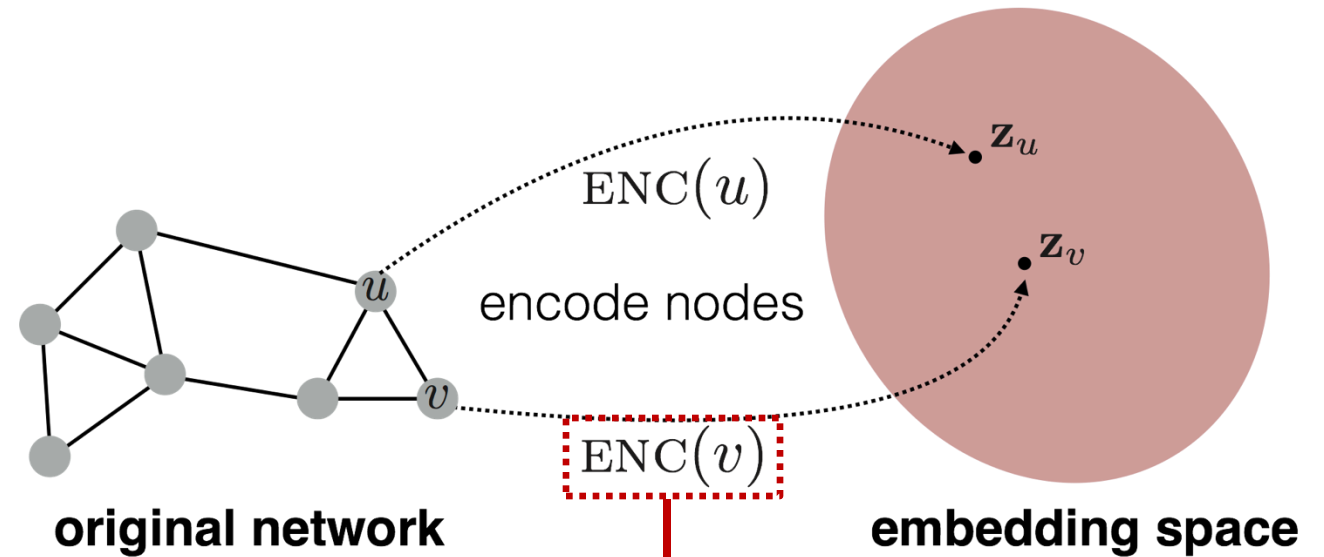
Policy making

- Graphs: *flexible* and *expressive*
- Graphs can **bridge interdisciplinary data**

Graph ML Tasks



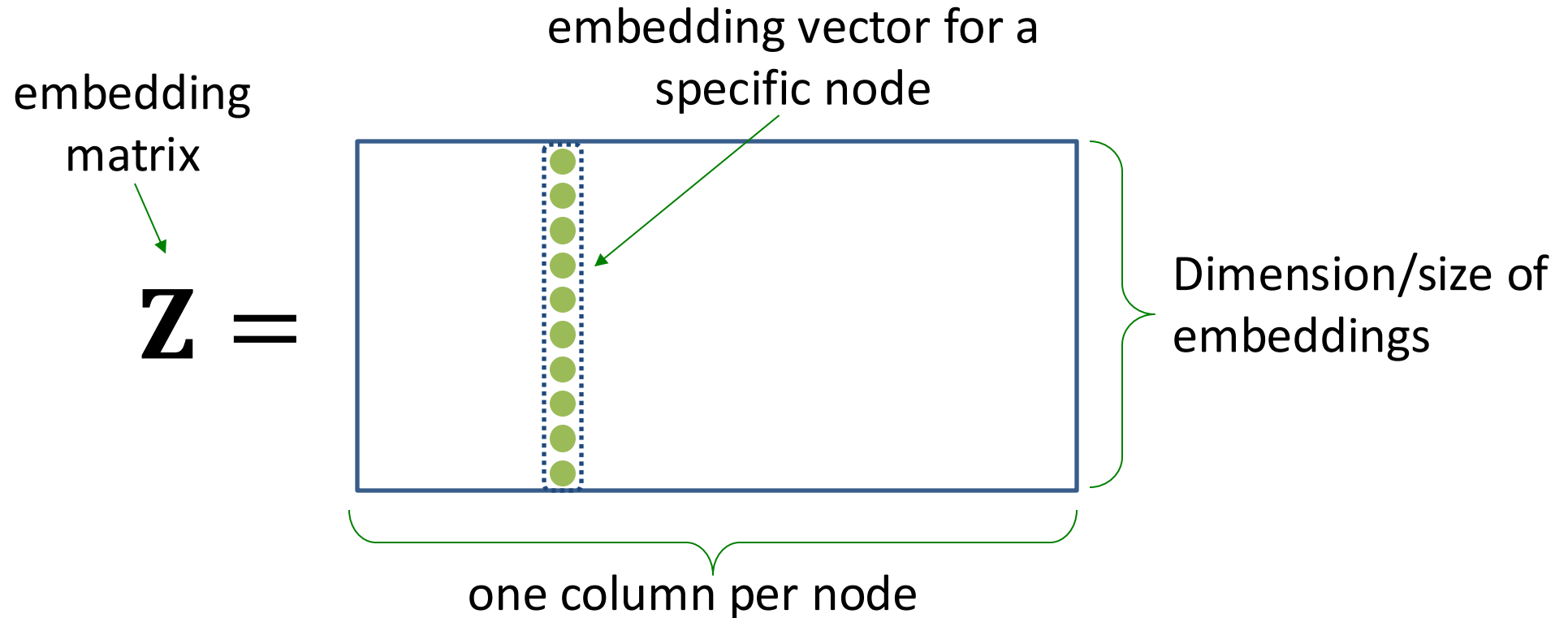
Key Idea: Node Embeddings



**Embedding Matrix,
Graph Neural Networks,**

“Shallow” Encoding

- Simplest encoding approach: **encoder is just an embedding-lookup**



A Single GNN Layer

- **Putting things together:**

- **(1) Message:** each node computes a message

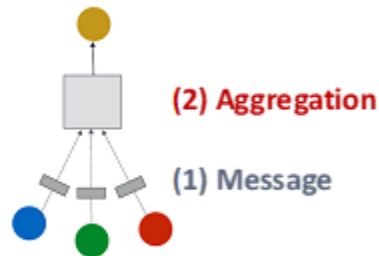
$$\mathbf{m}_u^{(l)} = \text{MSG}^{(l)} \left(\mathbf{h}_u^{(l-1)} \right), u \in \{N(v) \cup v\}$$

- **(2) Aggregation:** aggregate messages from neighbors

$$\mathbf{h}_v^{(l)} = \text{AGG}^{(l)} \left(\left\{ \mathbf{m}_u^{(l)}, u \in N(v) \right\}, \mathbf{m}_v^{(l)} \right)$$

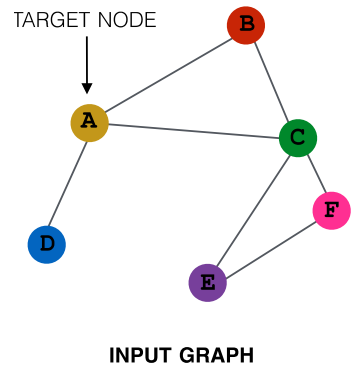
- **Nonlinearity (activation):** Adds expressiveness

- Often written as $\sigma(\cdot)$: $\text{ReLU}(\cdot)$, $\text{Sigmoid}(\cdot)$, ...
- Can be added to **message or aggregation**

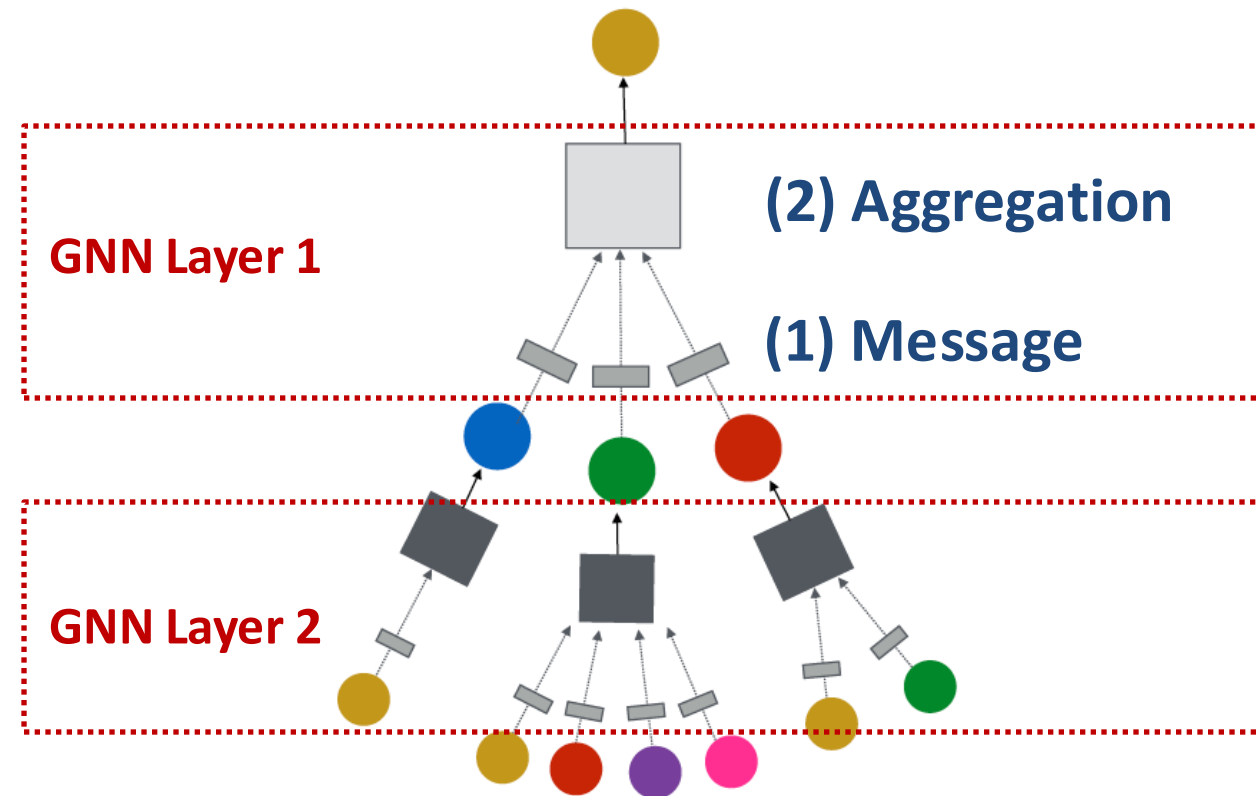


Recap: A General GNN Framework

(5) Learning objective

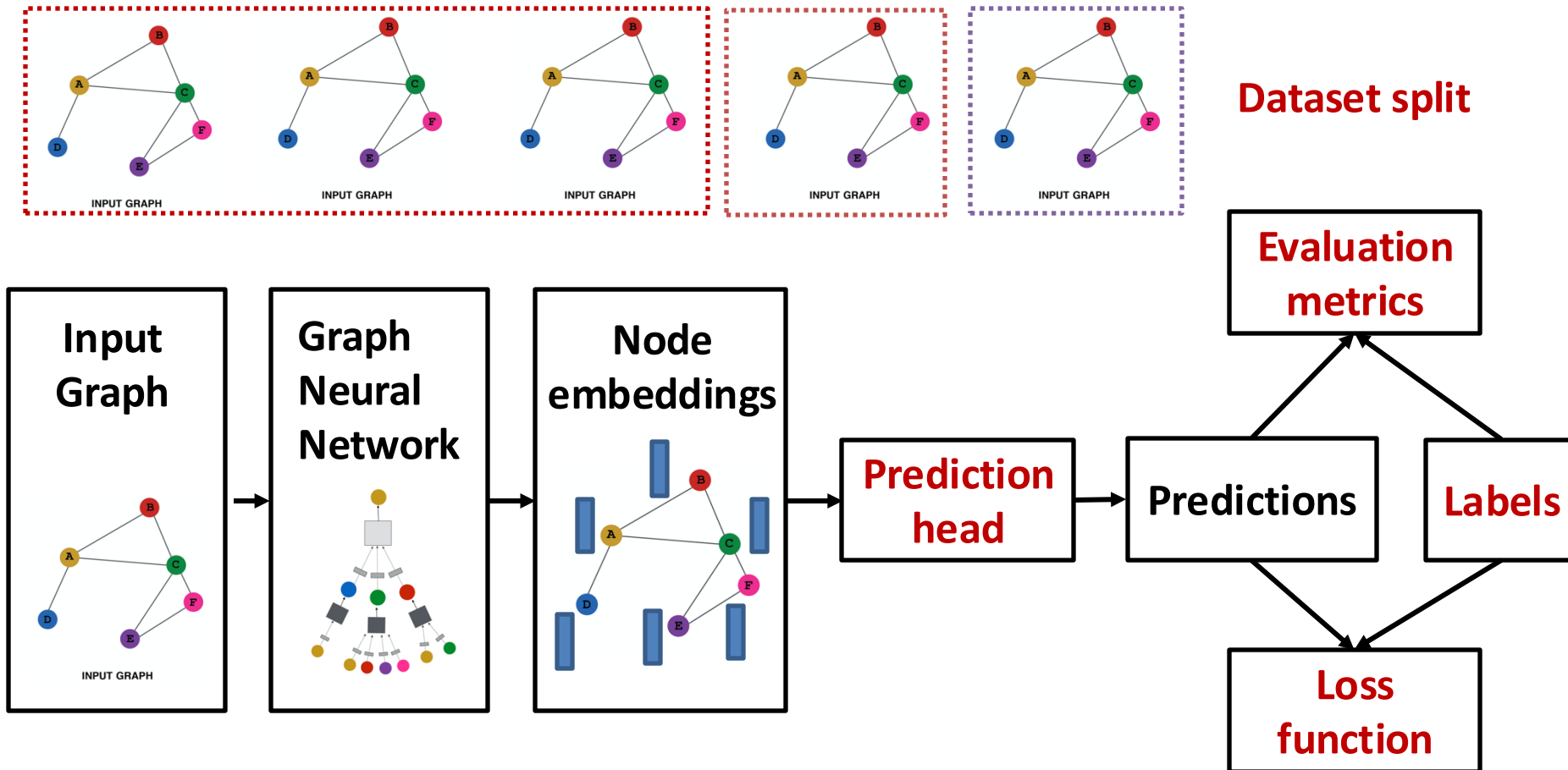


(3) Layer connectivity



(4) Graph augmentation

GNN Training Pipeline



Implementation resources:

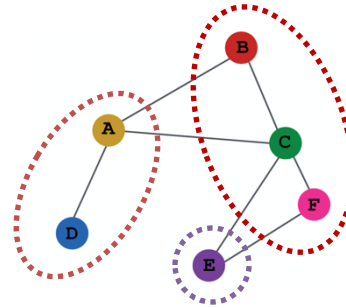
[DeepSNAP](#) provides core modules for this pipeline

[GraphGym](#) further implements the full pipeline to facilitate GNN design

Example: Node Classification

- **Transductive** node classification

- All the splits can observe the entire graph structure, but can only observe the labels of their respective nodes



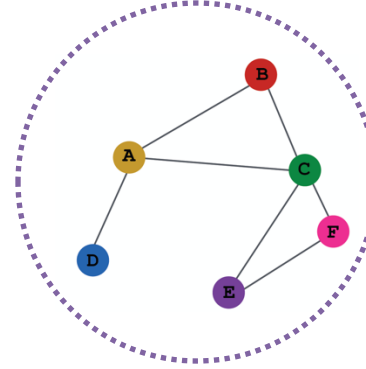
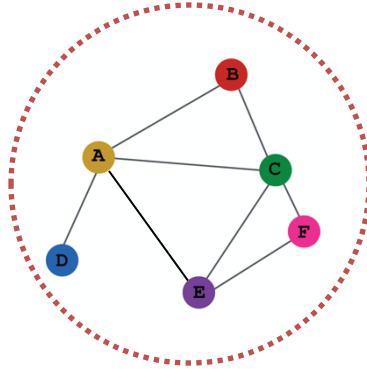
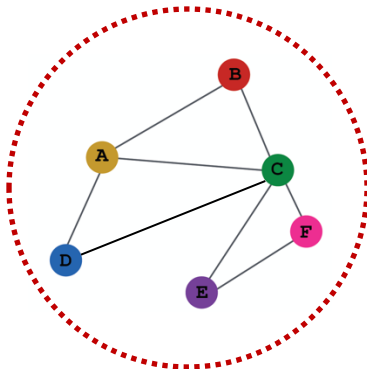
Training

Validation

Test

- **Inductive** node classification

- Suppose we have a dataset of 3 graphs
- Each split contains an independent graph



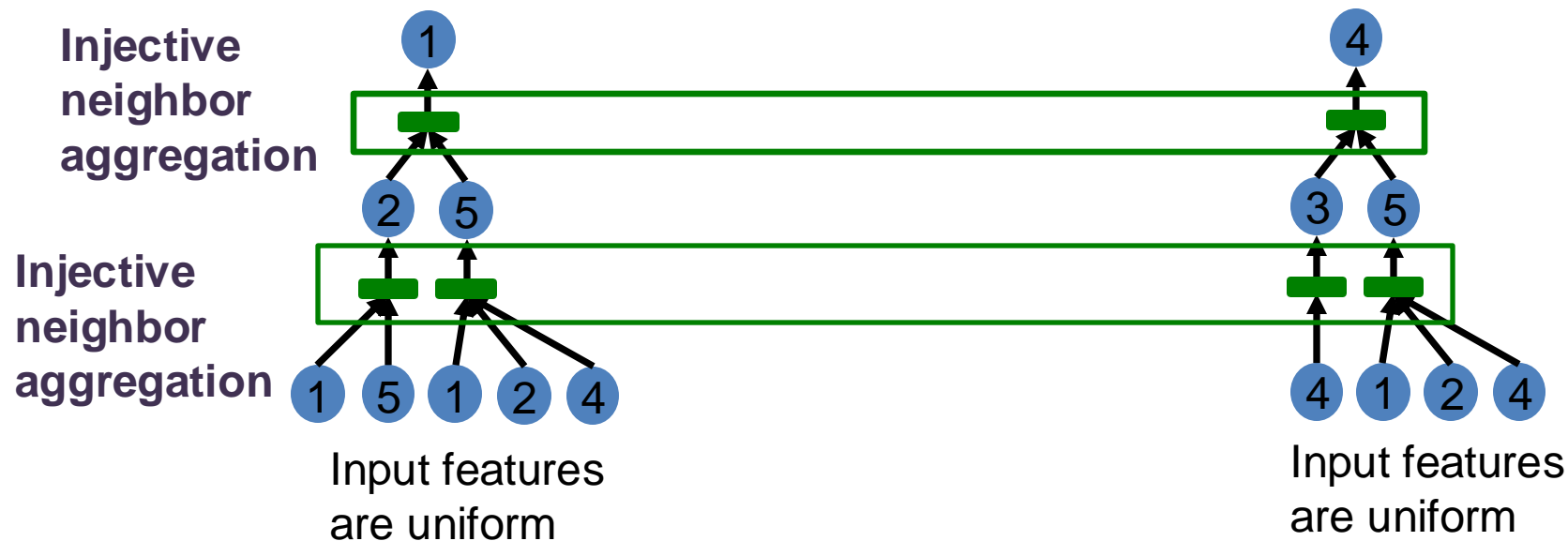
Training

Validation

Test

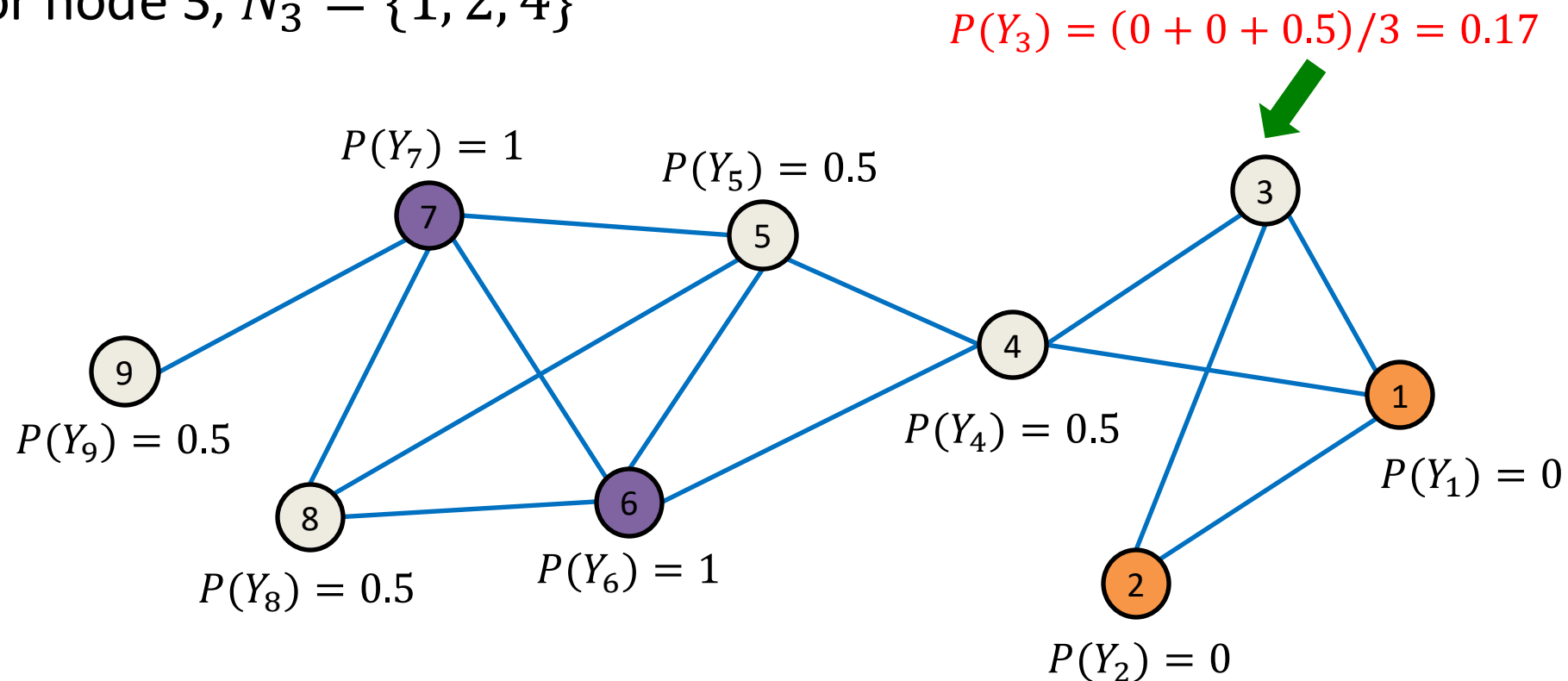
How Expressive is a GNN?

- In other words, most expressive GNN would use an **injective neighbor aggregation** function at each step.
 - Maps different neighbors to different embeddings.



Example: 1st Iteration, Update Node 3

- Update for the 1st Iteration:
 - For node 3, $N_3 = \{1, 2, 4\}$



GNN Design Space

Overall: A GNN design space

Intra-layer design

Batch Normalization	Dropout	Activation	Aggregation
True, False	False, 0.3, 0.6	RELU, PRELU, SWISH	MEAN, MAX, SUM

Inter-layer design

Layer connectivity	Pre-process layers	Message passing layers	Post-precess layer:
STACK, SKIP-SUM, SKIP-CAT	1, 2, 3	2, 4, 6, 8	1, 2, 3

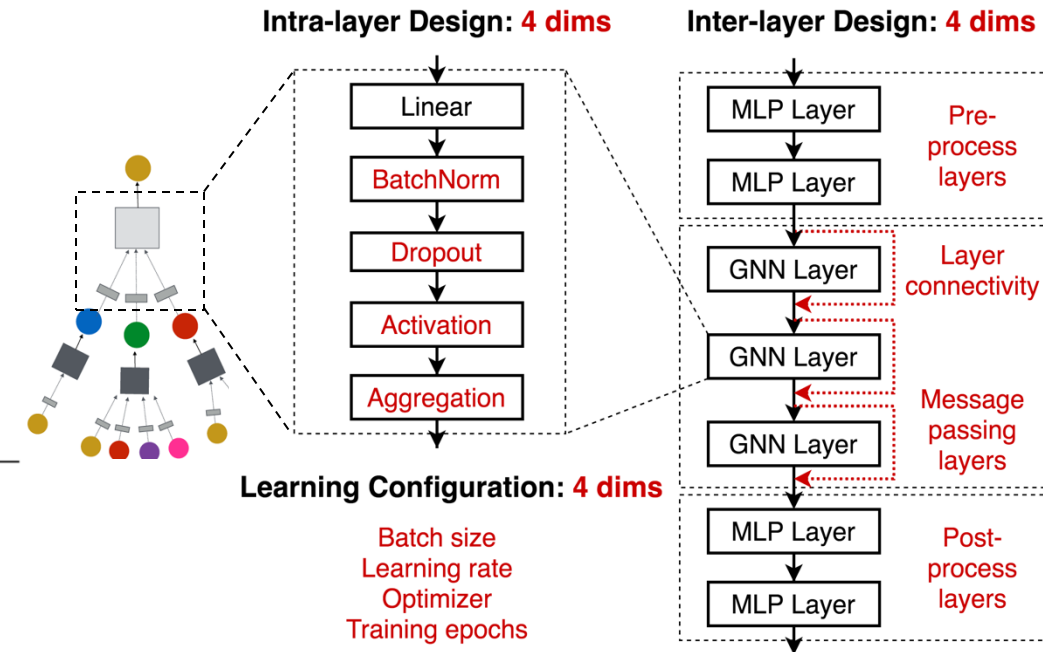
Learning configuration

Batch size	Learning rate	Optimizer	Training epochs
16, 32, 64	0.1, 0.01, 0.001	SGD, ADAM	100, 200, 400

In total: 315K possible designs

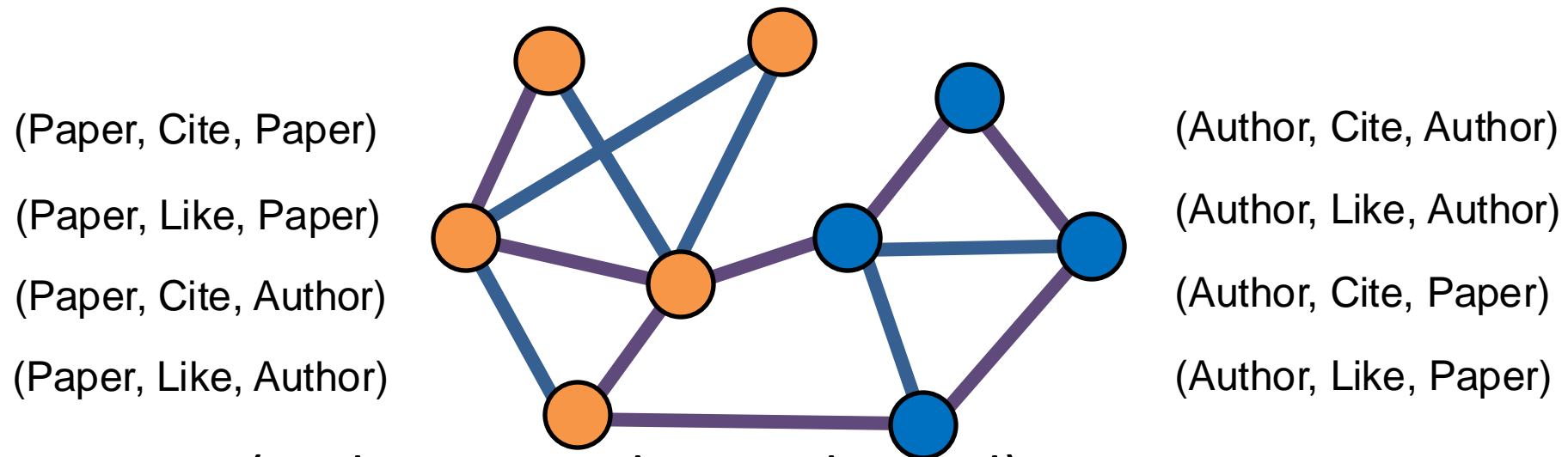
Purpose:

- We don't want to (and we cannot) cover all the possible designs
- A mindset transition:** We want to demonstrate that **studying a design space is more effective than studying individual GNN designs**



Heterogeneous Graphs: Motivation

8 possible relation types!



Relation types: (node_start, edge, node_end)

- We use **relation type to describe an edge** (as opposed to edge type)
- Relation type better captures the interaction between nodes and edges

TransE

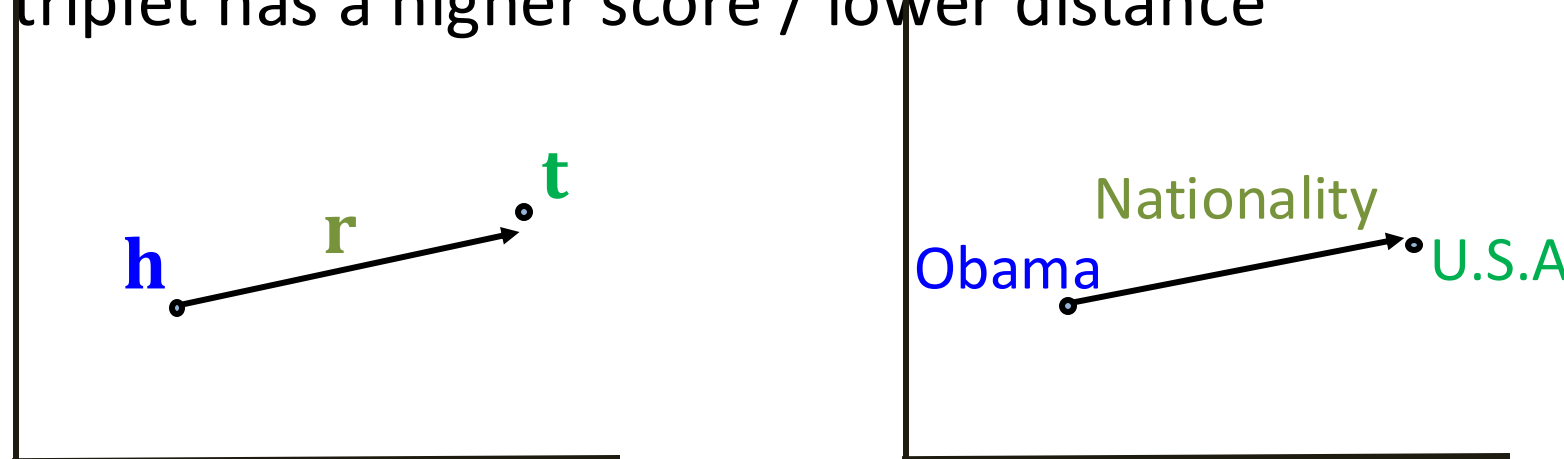
- **Intuition: Translation**

For a triplet (h, r, t) , let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ be embedding vectors.

- **TransE:** $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ if the given link exists else $\mathbf{h} + \mathbf{r} \neq \mathbf{t}$

Entity scoring function: $f_r(h, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||$

- A valid triplet has a higher score / lower distance



embedding
vectors will
appear in
boldface

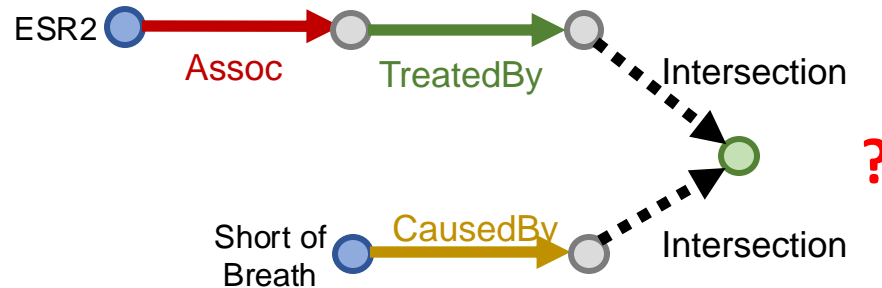
Embed with Box Embedding

“What is the drug that causes Short of Breath and treats disease associated with protein ESR2?”

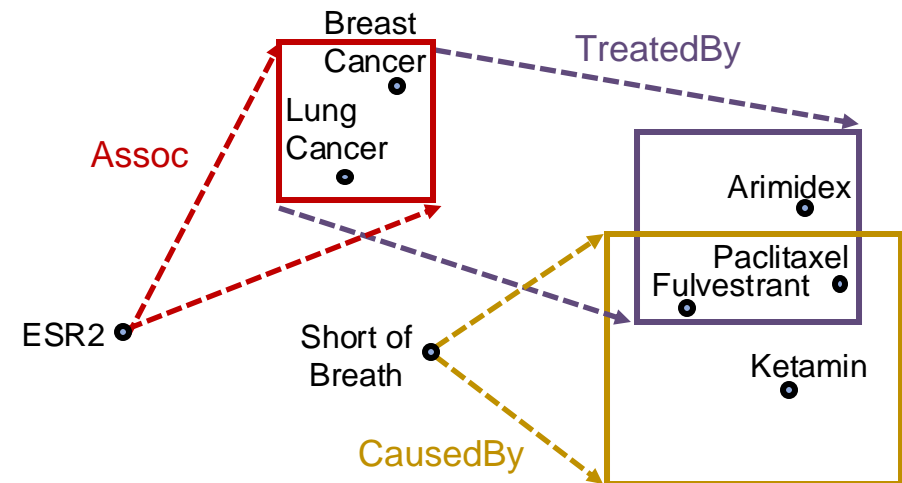
$((e:ESR2, (r:Assoc, r:TreatedBy)), (e:Short\ of\ Breath, (r:CausedBy)))$

- How do we take intersection of boxes?

Query Plan

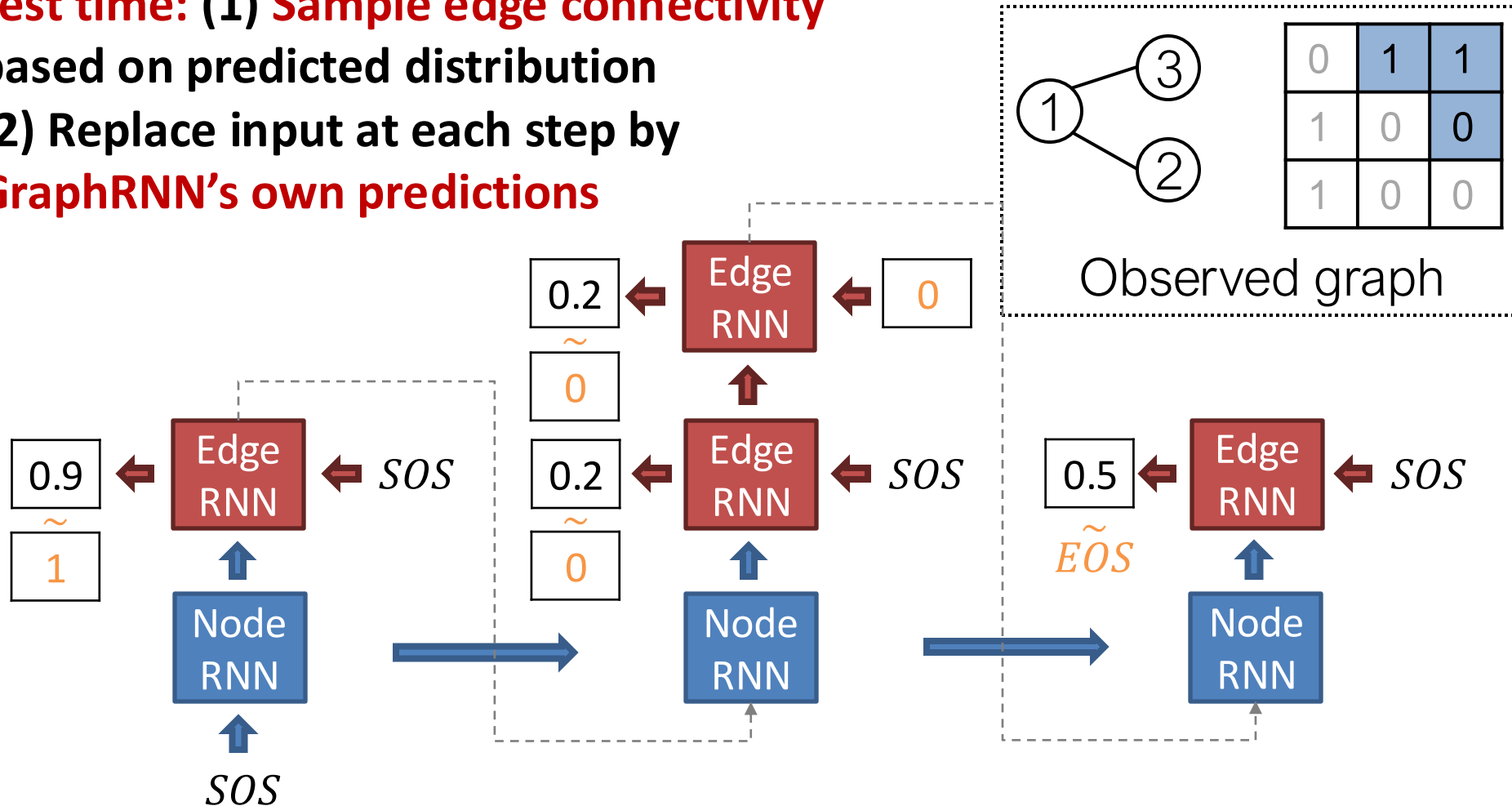


Embedding Space



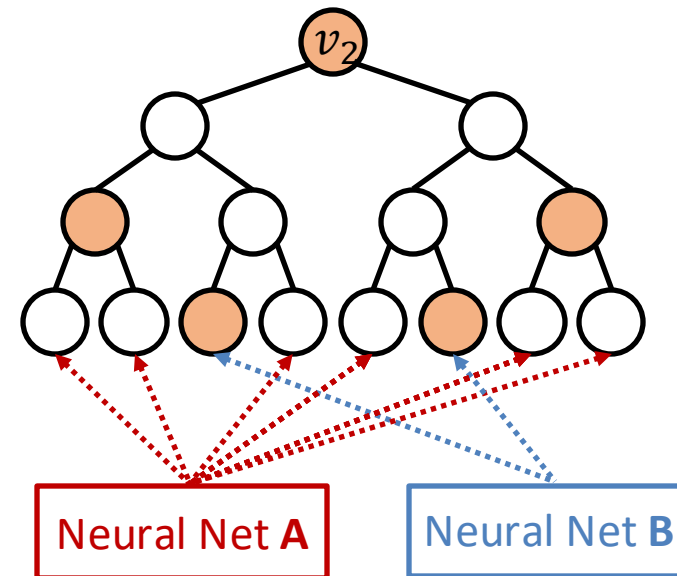
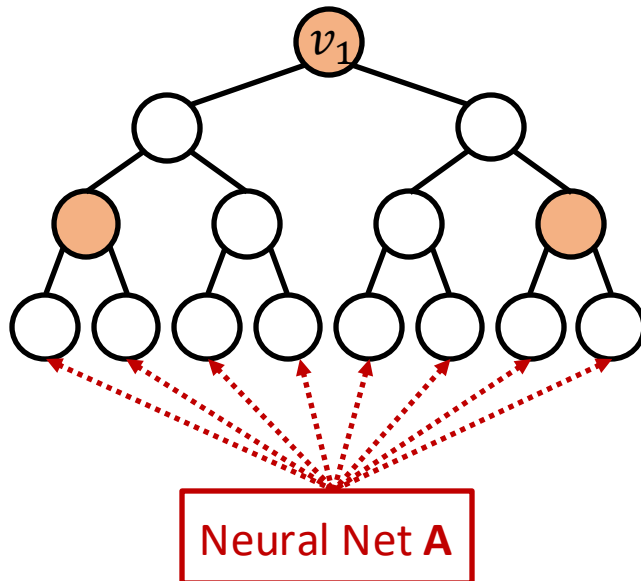
Put Things Together: Test

Test time: (1) Sample edge connectivity based on predicted distribution
(2) Replace input at each step by GraphRNN's own predictions



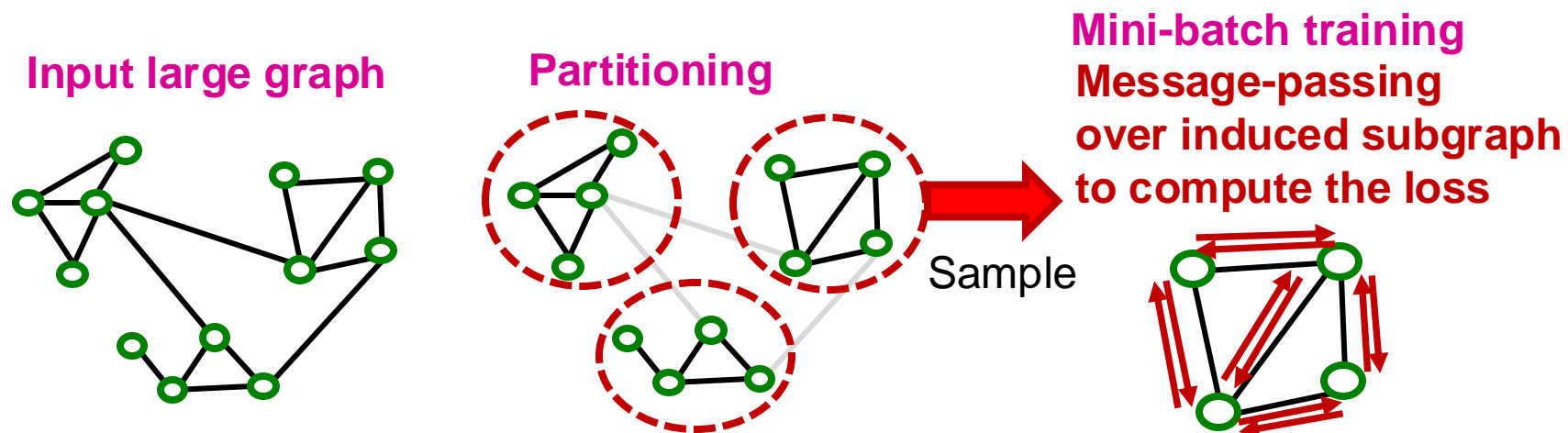
Identity-aware GNN

- Why does heterogenous message passing work:
 - Suppose two nodes v_1, v_2 have the same computational graph structure, but have different node colorings
 - Since we will apply different neural network for embedding computation, their embeddings will be different



Cluster-GCN: Overview

- We first introduce “vanilla” Cluster-GCN.
- Cluster-GCN consists of two steps:
 - **Pre-processing**: Given a large graph, partition it into groups of nodes (i.e., subgraphs).
 - **Mini-batch training**: Sample one node group at a time. Apply GNN’s message passing over the **induced subgraph**.



How to inject PE/SE

- Inject local PE/SE with node inputs and treat relative PE/SE as additional attention bias
 - Example: Graphormer

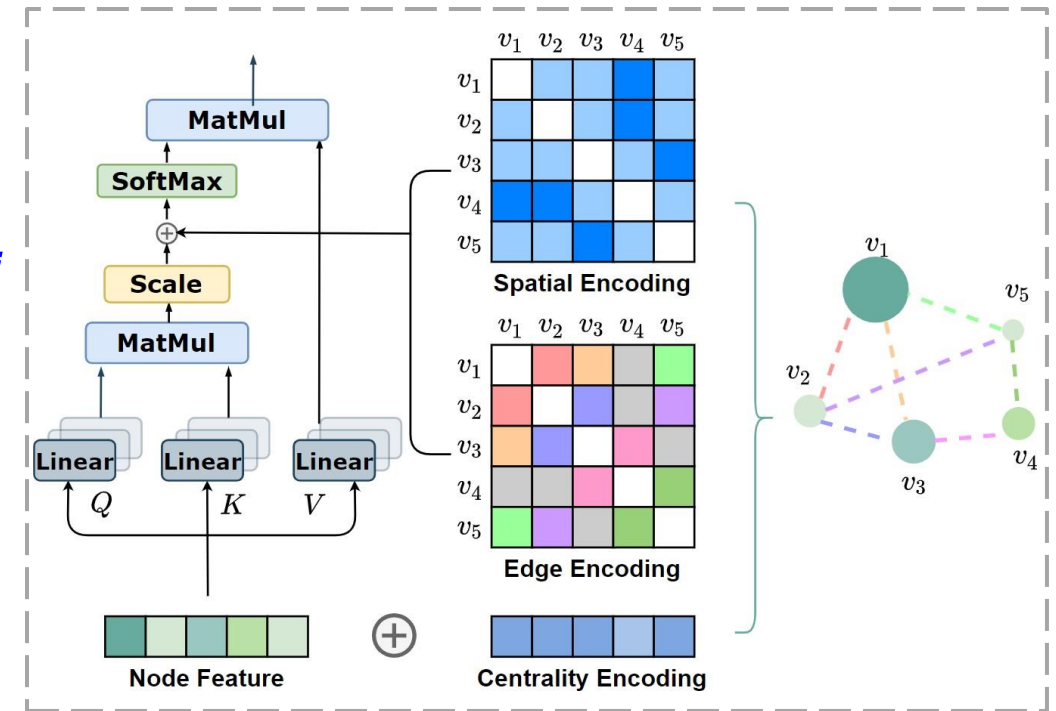
$$\text{Attention: } e_{ij} = \frac{(h_i W_q)(h_j W_k)^T}{\sqrt{d}} + b_{\phi(v_i, v_j)} + c_{ij}$$

Spatial Encoding:

Shortest path between v_i, v_j

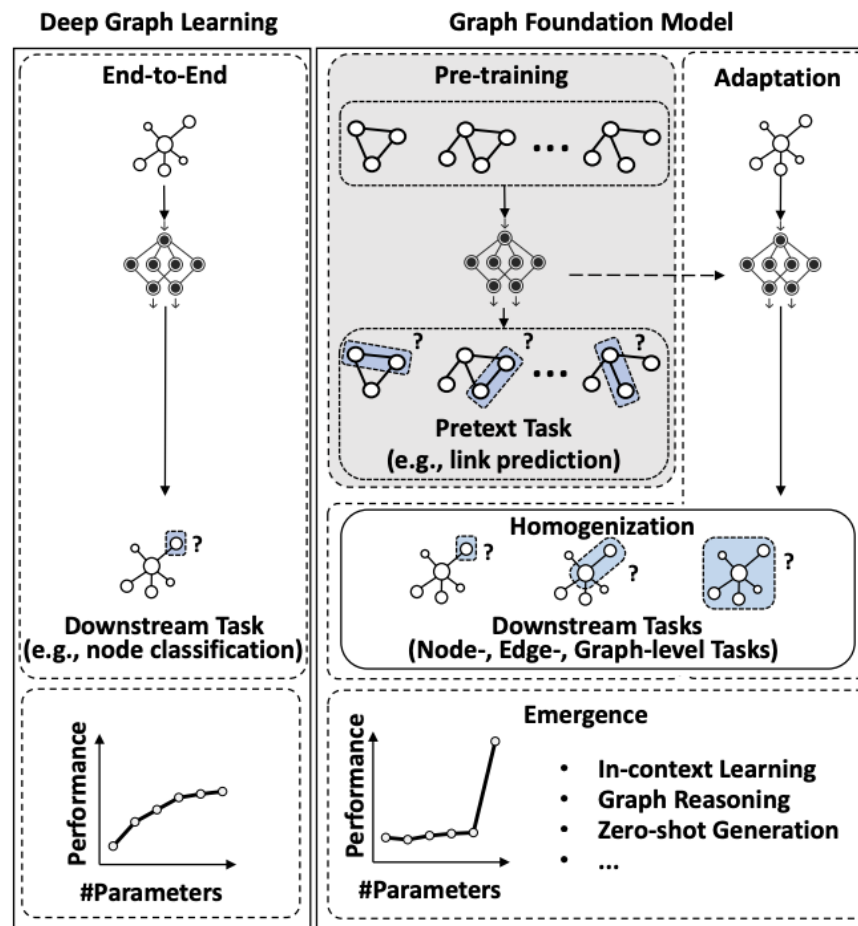
Edge Encoding:

Average all edge features along the shortest path between v_i, v_j , $c_{ij} = \frac{1}{N} \sum_{e \in SP(i,j)} x_e w_e$



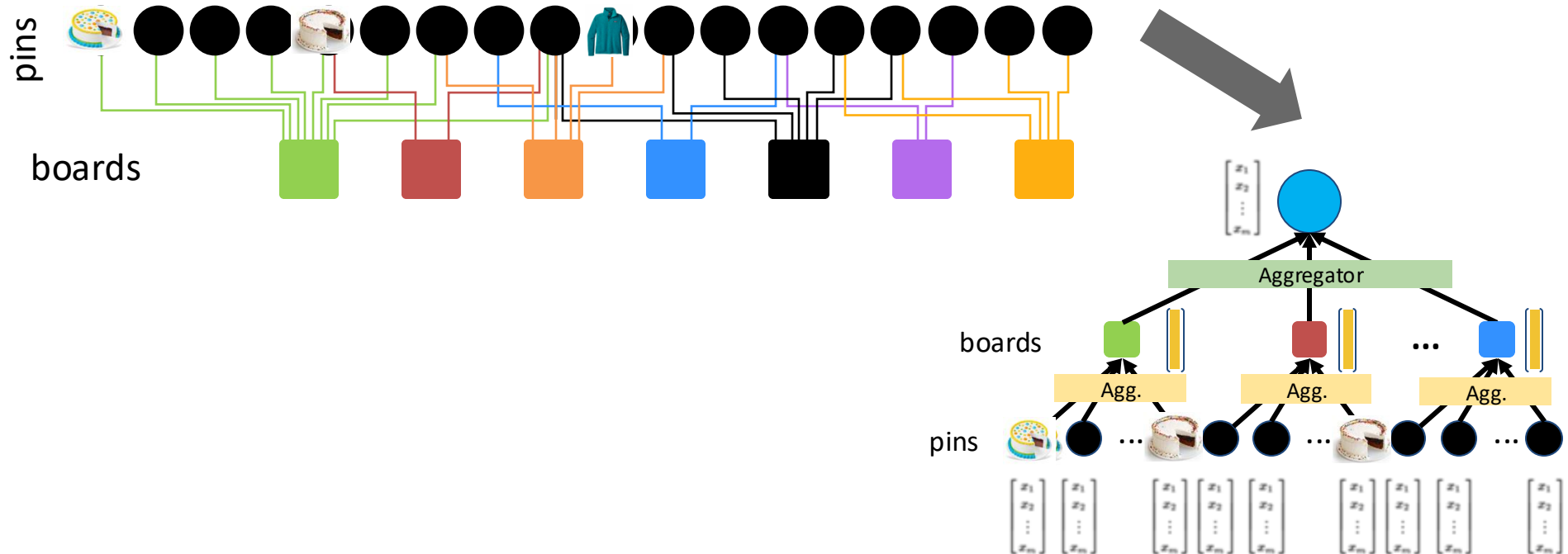
Graph Foundation Models

- A graph foundation model (GFM) is a model pre-trained on **extensive graph data**, adapted for **diverse downstream graph tasks**.



PinSAGE: Graph Neural Network

- Graph has tens of billions of nodes and edges
- Further resolves embeddings across the Pinterest graph

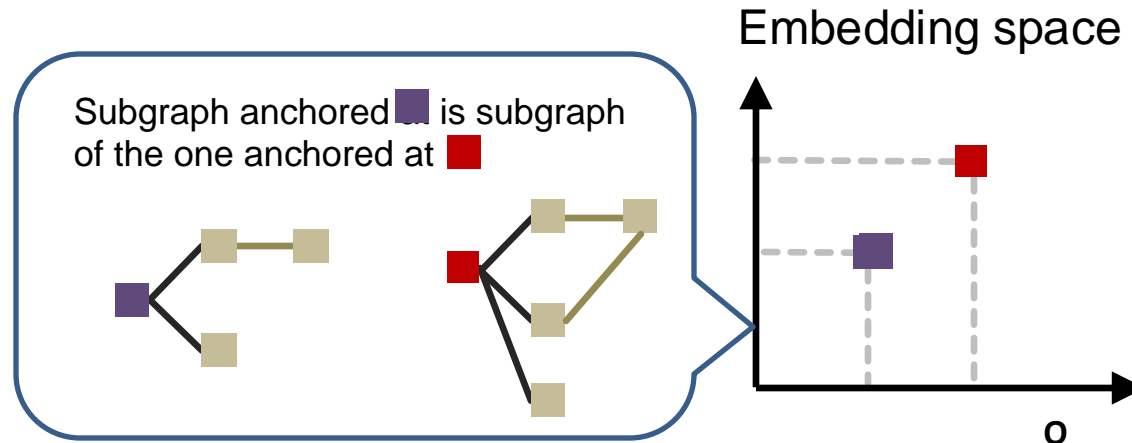


Order Constraint (2)

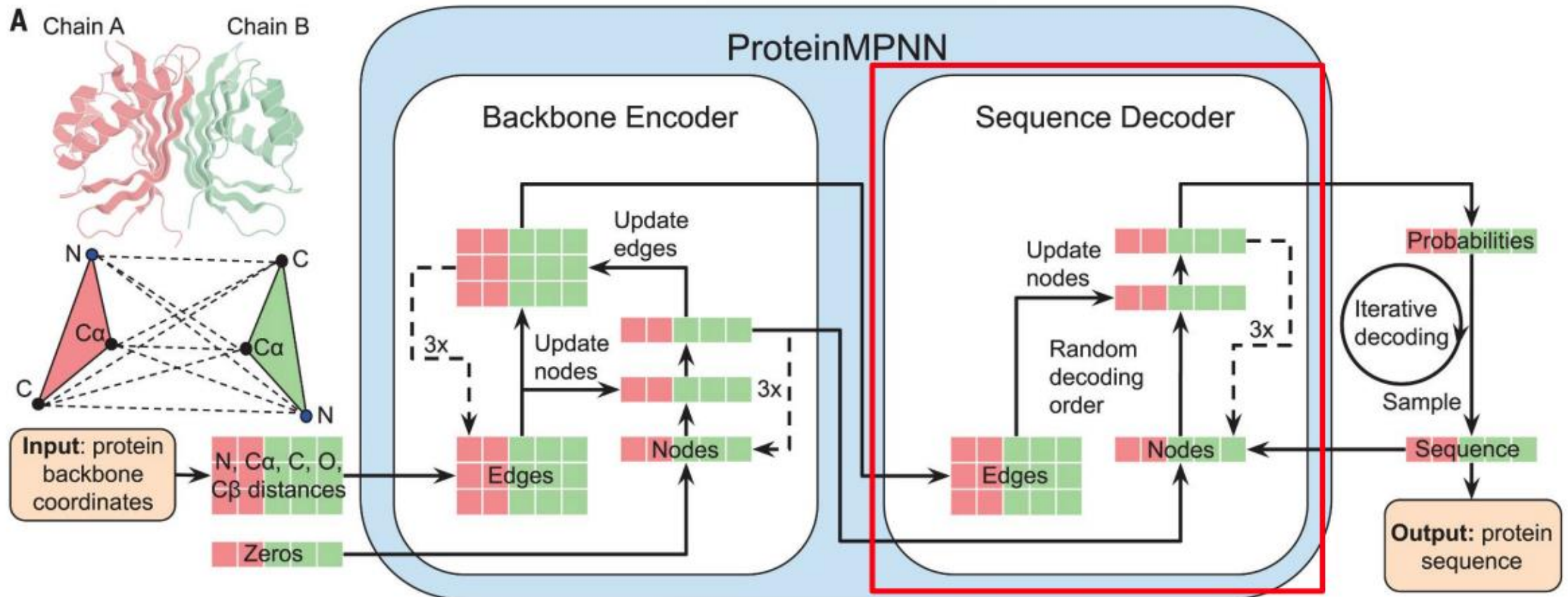
- We specify the order constraint to ensure that the subgraph properties are preserved in the order embedding space

$$\forall_{i=1}^D z_q[i] \leq z_t[i] \quad \text{iff} \quad G_Q \subseteq G_T \quad \text{trained with max-margin loss}$$

Query embedding Target embedding Subgraph Relation Embedding dimension



Autoregressive Decoder in ProteinMPNN



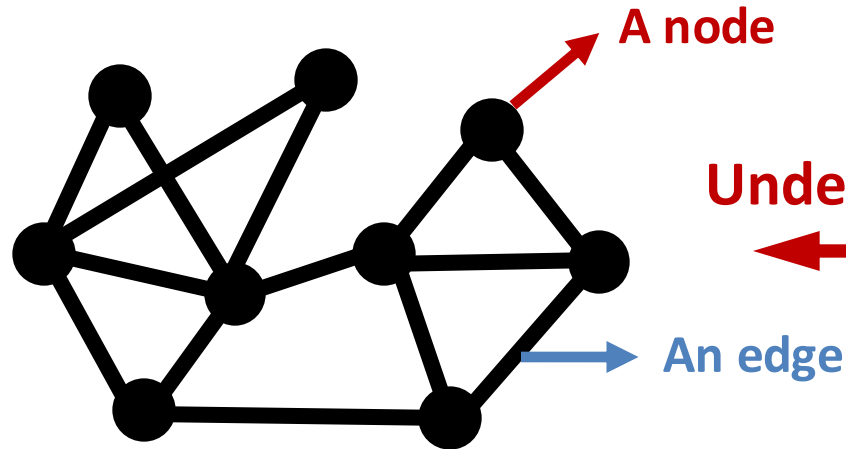
The Bottom Line

- There is exciting relational structure in many many real-world problems
- Identifying and harnessing this relational structure leads to better AI



Interconnected world

Represent



Graph-structured data

Understand



AI agent

Conclusion

Training for AI research

The 5 Questions for A Good Paper

(Originally for writing paper introduction)

- What is the problem?
- Why is it interesting and important?
- Why is it hard? (E.g., why do naive approaches fail?)
- Why hasn't it been solved before? (Or, what's wrong with previous proposed solutions? How does mine differ?)
- What are the key components of my approach and results?
Also include any specific limitations.

What's Next?

- **Review and Response (Due on Dec 8)**
 - Everyone will be **assigned 2 papers to review** by this weekend.
 - Each group is required to **participate in the public discussion** of the papers. We will discuss general principles and practical strategies today.
 - Review and response will count towards **10%** of your final grade. We will evaluate both your reviews and your responses.

What's Next?

- **Presentation** (Due on **Dec 4 & Dec 6**)
 - Each group is required to prepare slides for the presentation.
 - We will have **5 minutes** for presentation and **3 minutes** for comments.
 - TA will assign each group into the Wednesday or Friday session by Dec 1.
 - Presentation will count towards **10%** of your final grade.

Conclusion

Review & Response

Reviewer Guide

- **An outline of the main reviewer tasks**
- **Step-by-step reviewing instructions**
- **Review Examples**

High Quality Review

- **1 Summary:** Provide a brief, accurate summary of the paper, emphasizing its key contributions, methodology, and findings.
- Example: *"This paper proposes a novel method for optimizing transformer architectures using a learned attention pruning mechanism. The authors demonstrate improvements in both computational efficiency and downstream performance across three benchmark datasets."*

High Quality Review

- **2 Strengths:** Highlight the paper's major contributions and strengths in detail.
- Example: *"The approach is innovative and addresses a significant problem in transformer scalability. The experimental results are compelling, with a 20% reduction in computational cost and consistent performance gains."*

High Quality Review

- **3 Weaknesses:** Clearly state limitations or potential issues, such as incomplete comparisons, insufficient analysis, or unclear writing.
- Example: *"The method lacks an ablation study to disentangle the effects of the pruning mechanism from other components of the model."*

High Quality Review

- **4 Constructive Feedback:** Provide **actionable** suggestions for improvement.
- Example: *"Adding a comparison with recent pruning methods like XYZ et al. (2023) would strengthen the claims. Also, explaining why Dataset A was chosen over other common benchmarks would improve clarity."*

High Quality Review

- **Detailed and Specific Comments:** Avoid vague statements. Back your claims with evidence or examples.
- Weak: *"The experiments are insufficient."*
- Strong: *"The experiments focus only on synthetic data. Real-world datasets like ABC would provide stronger evidence for the method's generalizability."*

High Quality Rebuttal Tip

- Think about yourself as an LLM agent
- **“RAG”**
- Retrieve evidence
 - From your paper
 - Line xxx to xxx, Figure xxx, Table xxx
 - From famous paper
 - Relevant works, famous works
- Action
 - Improve your writing
 - Add new results

Conclusion

Open-ended Discussion - AMA