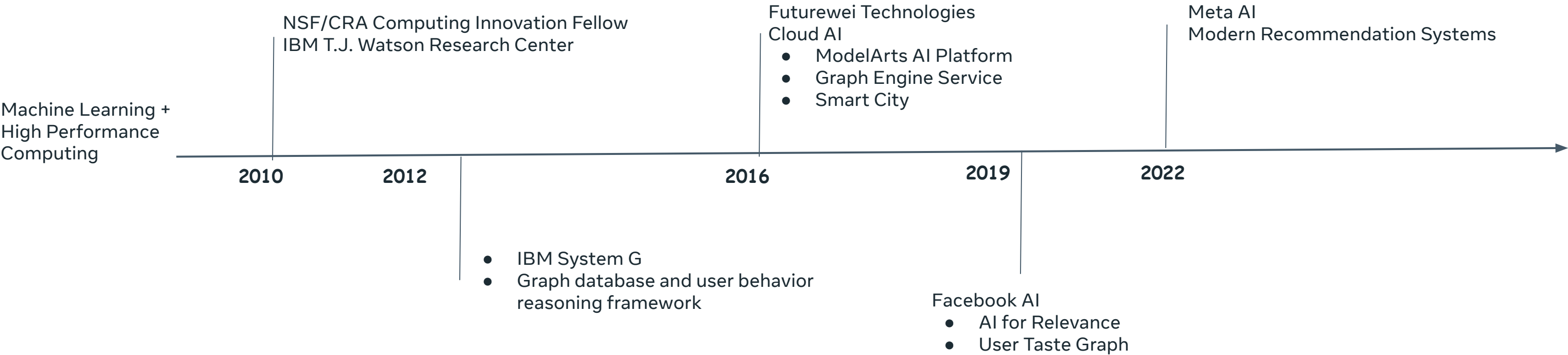


Machine Learning on Graph for Recommendation and more

Yinglong Xia
MRS, Meta

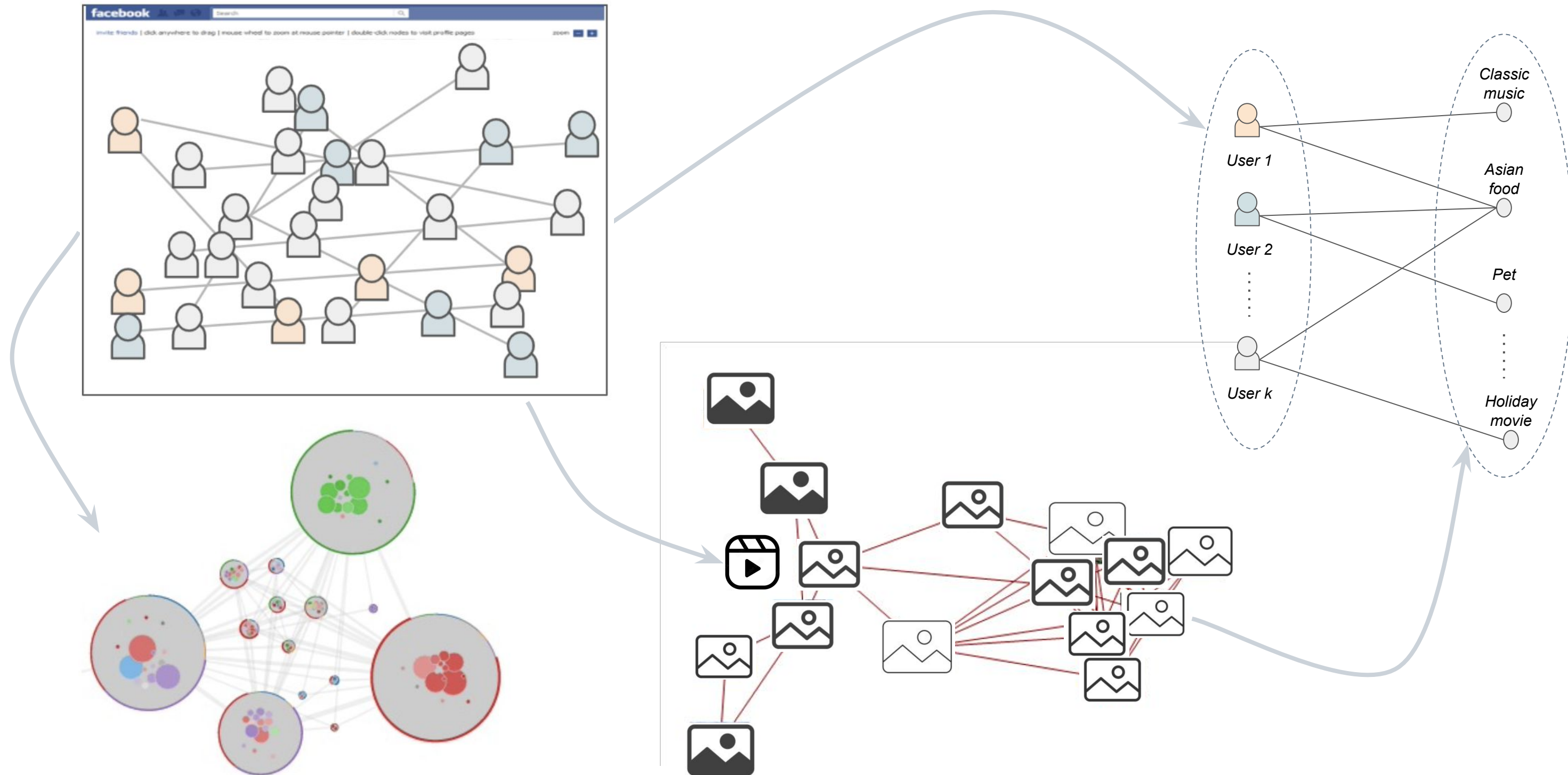
About me



Agenda

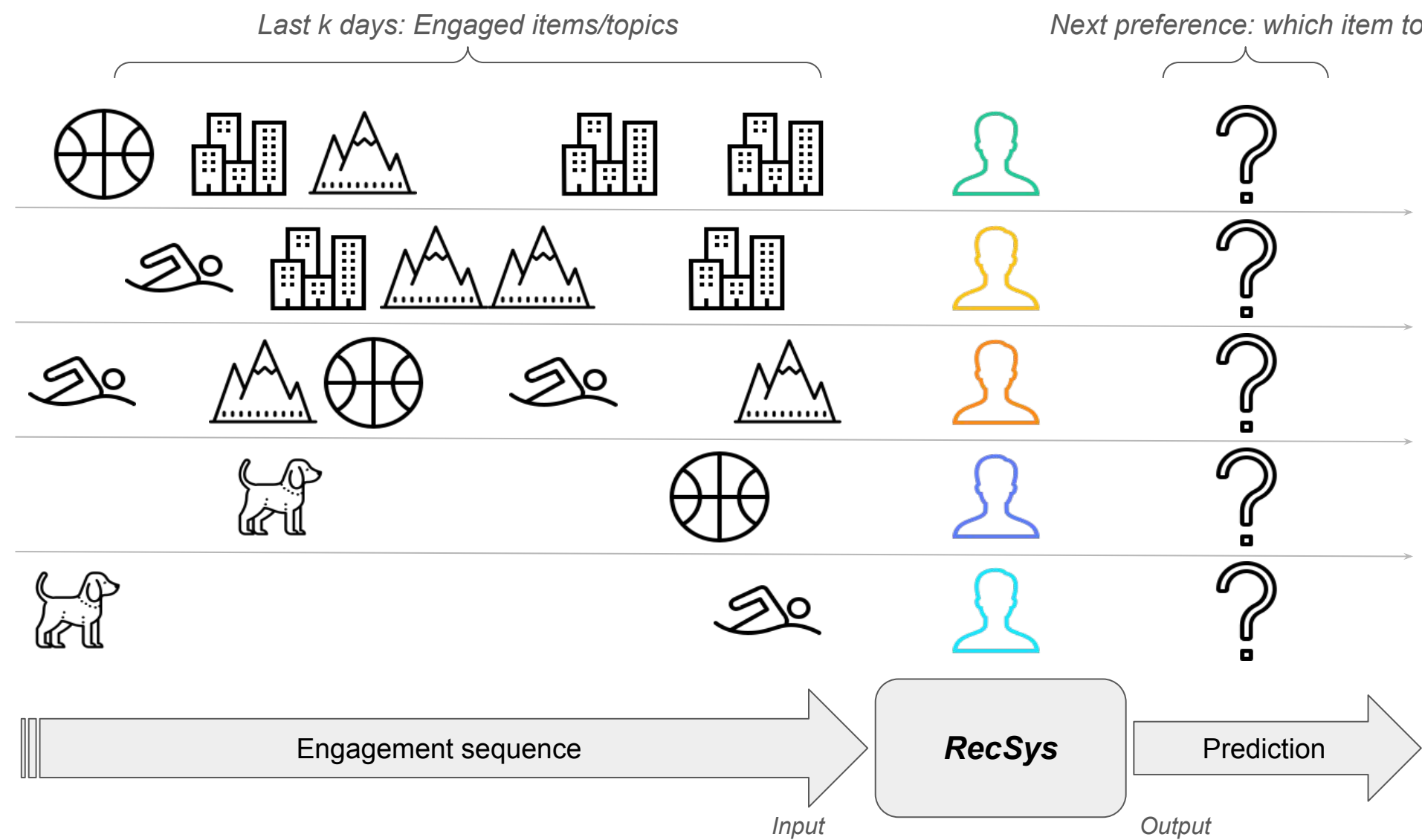
- 01 **Graph in RecSys**
- 02 Graph sparse nodes in recommendation
- 03 GNN expressiveness: depth, scope, & MoE
- 04 Conclusion & Opportunities

Graph is everywhere

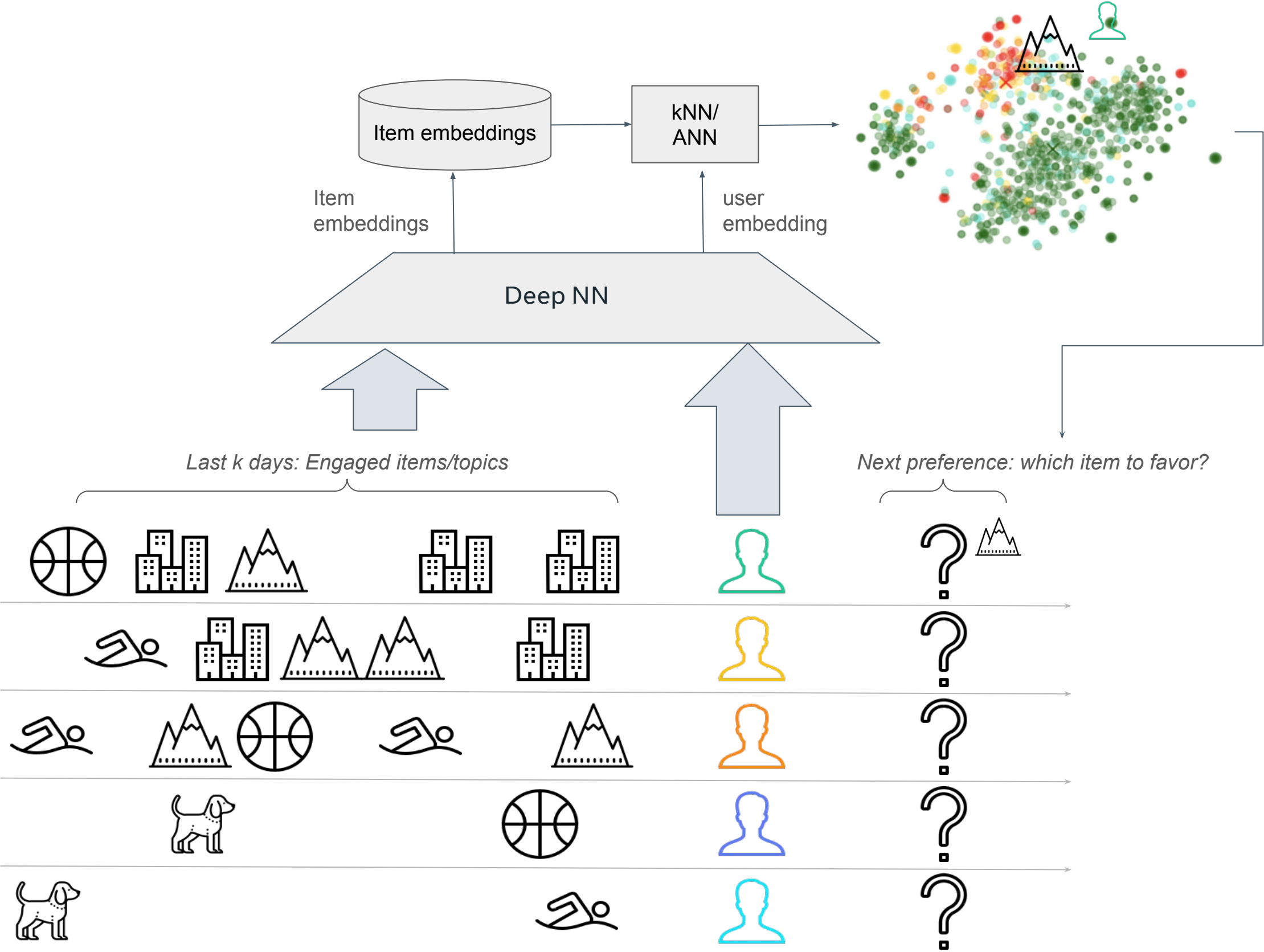


Recommendation system

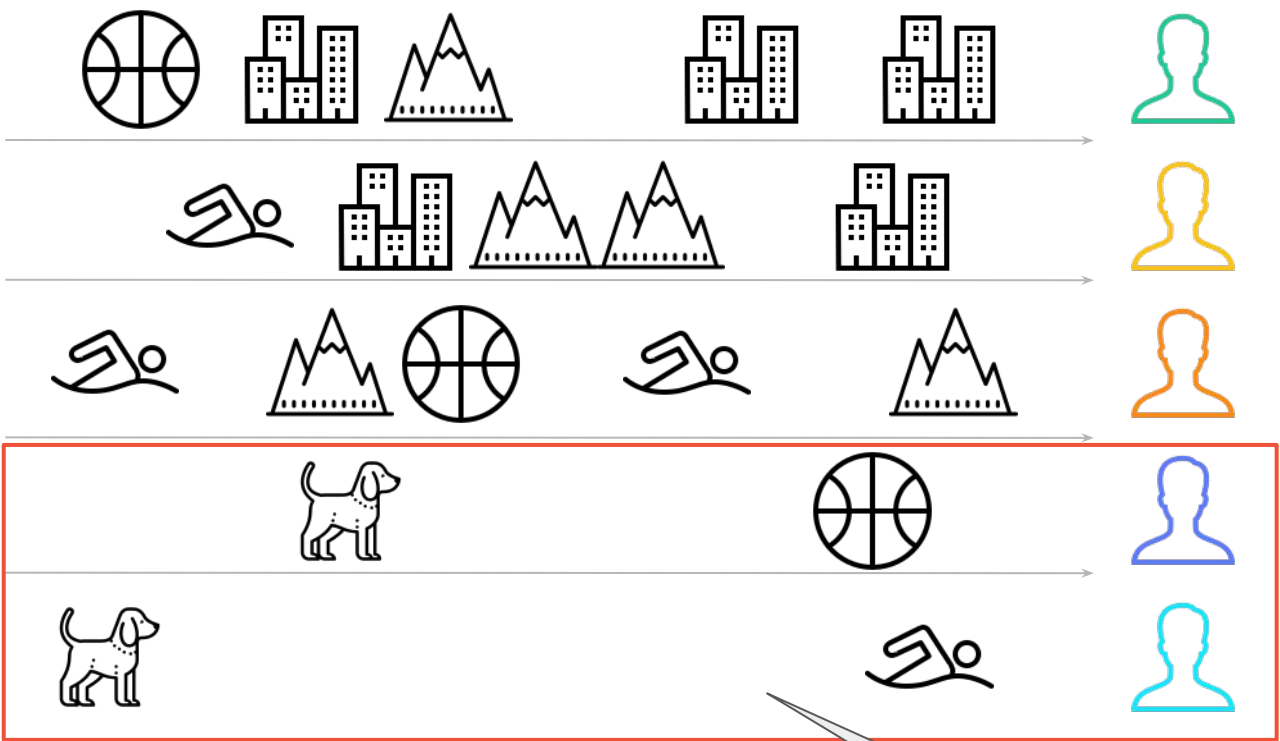
A recsys is a subclass of information filtering system that provides suggestions for items that are most pertinent to a particular user.



01 Graph in RecSys

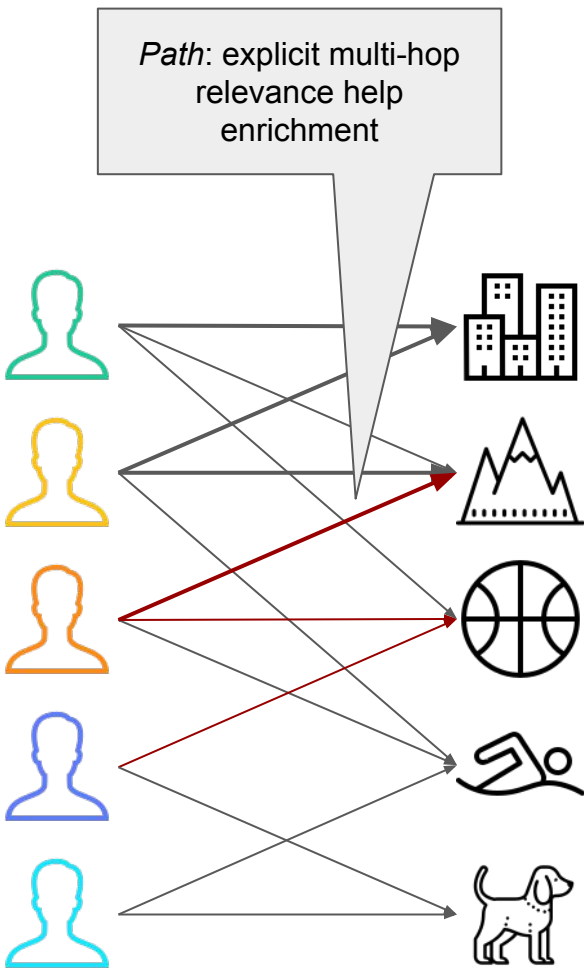
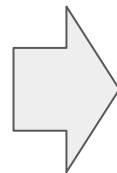


Sequence vs Graph

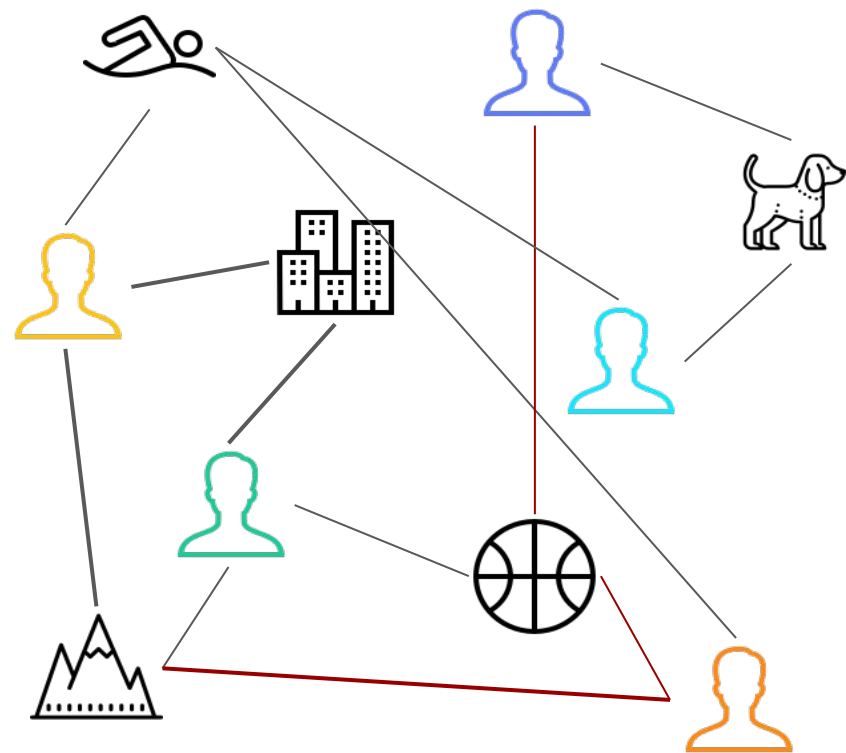


Sparse/noisy input sequence results in untrustworthy prediction

Lightweight users



User-item engagement bipartite graph



User-item heterogeneous graph

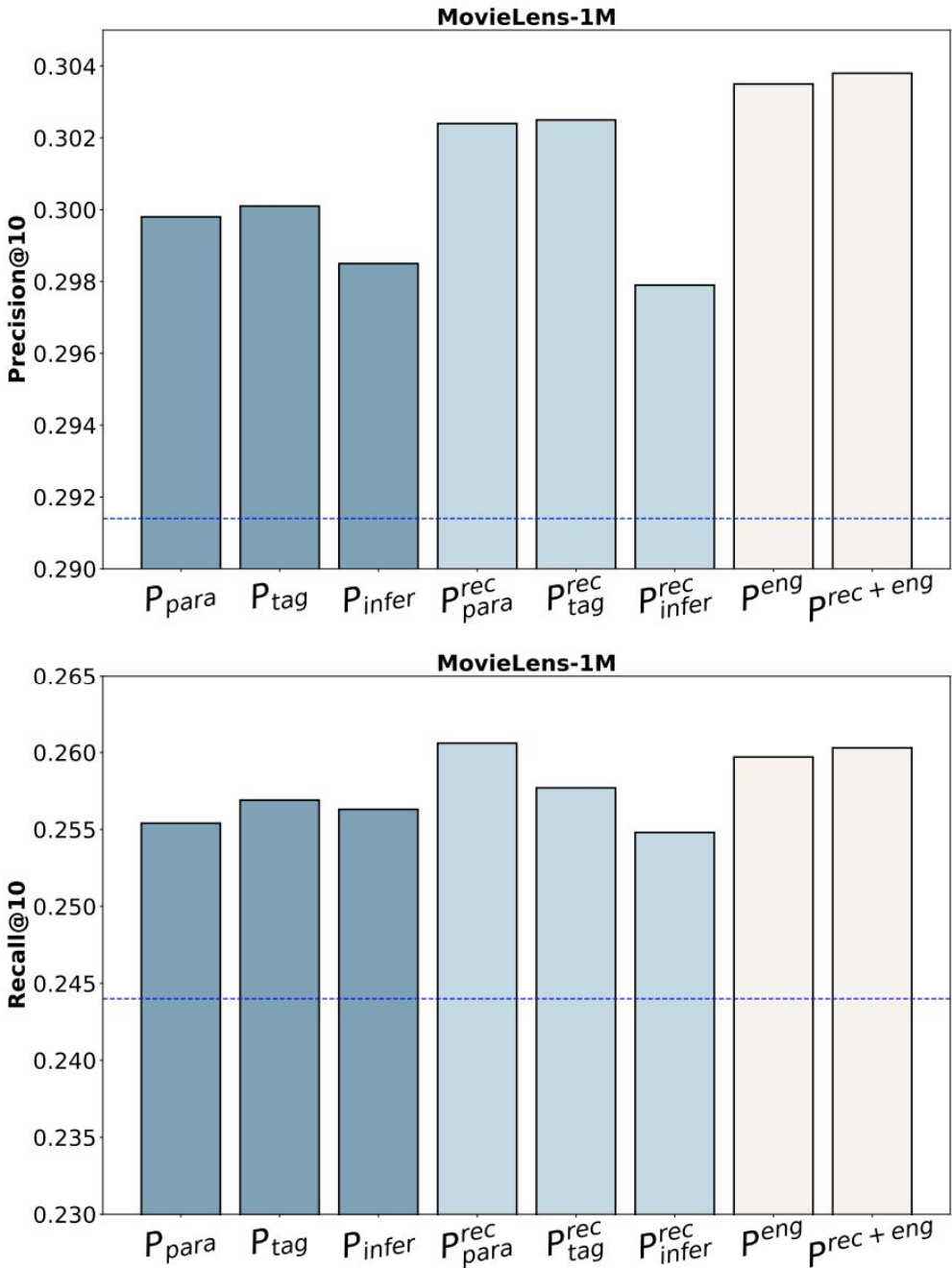
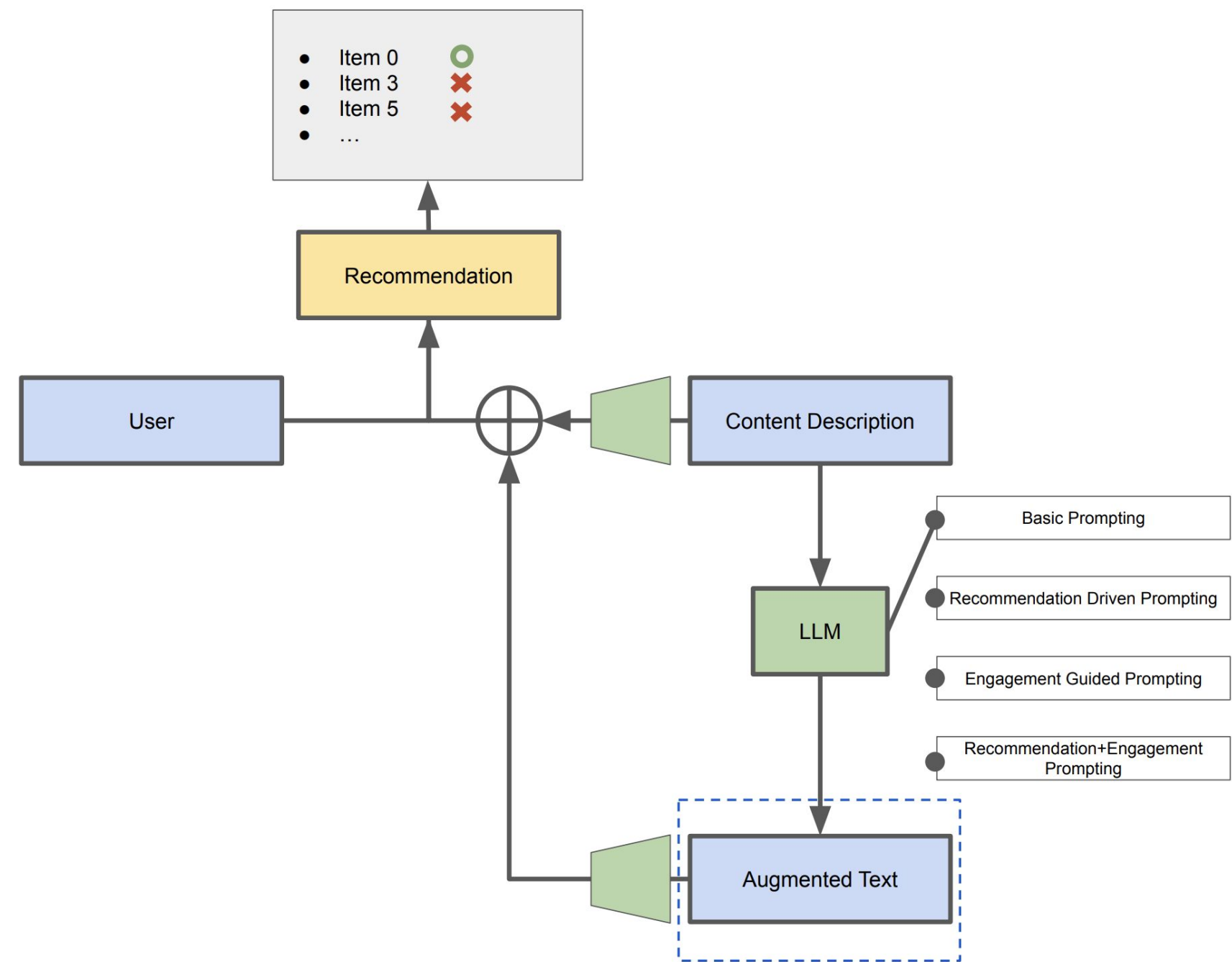
ML101: "Garbage in, garbage out" (GIGO)

- RecSys prediction counts on user engagement history
- Cold start users have limited & noisy engagement history, if not unavailable
- How can we enrich user signals by leveraging latent relevance?

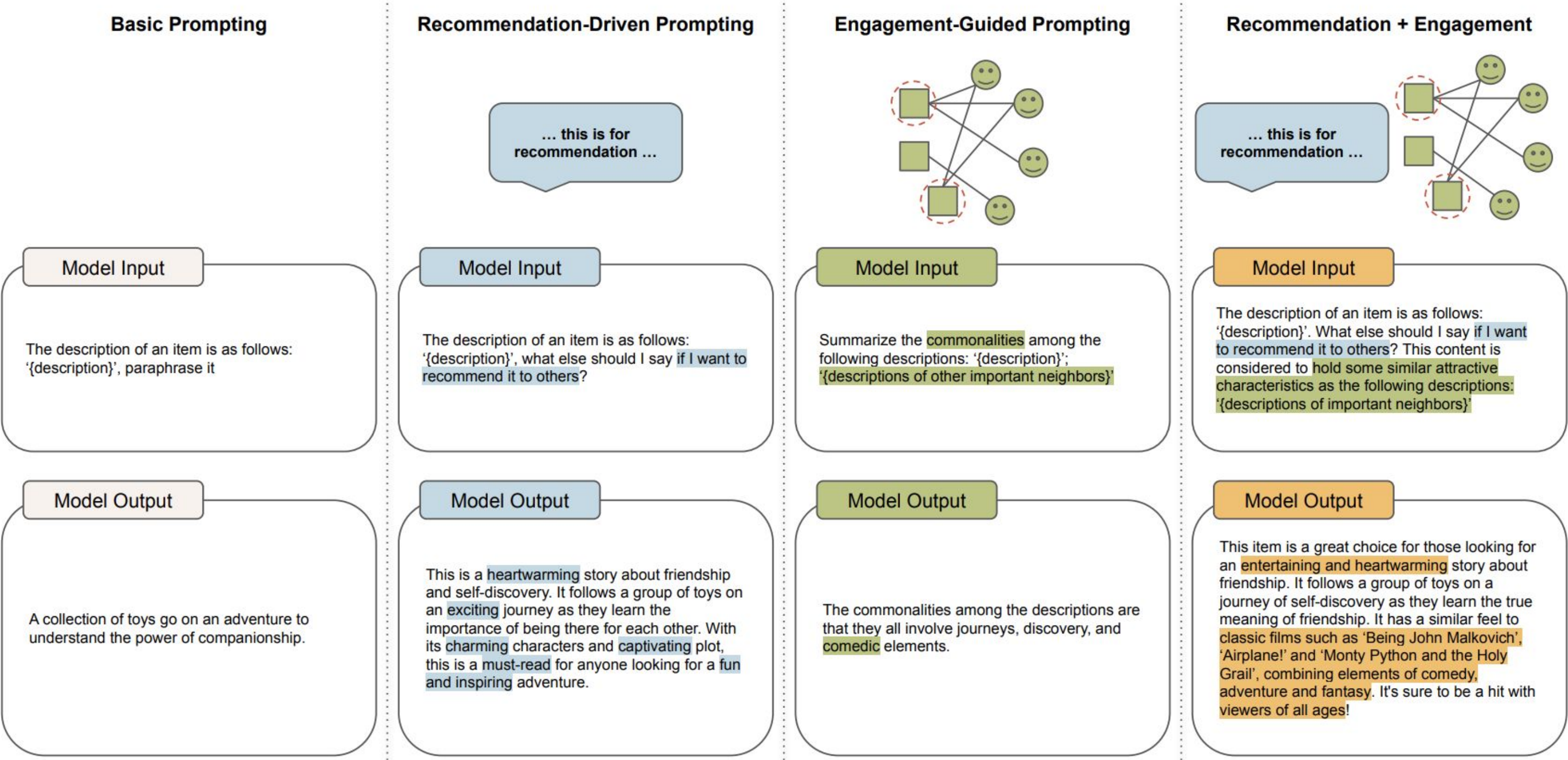
Graph meets LLM in RecSys

H. Zeng, Y. Xia, et al, LLM-Rec: Personalized Recommendation via Prompting Large Language Models, NAACL 2024

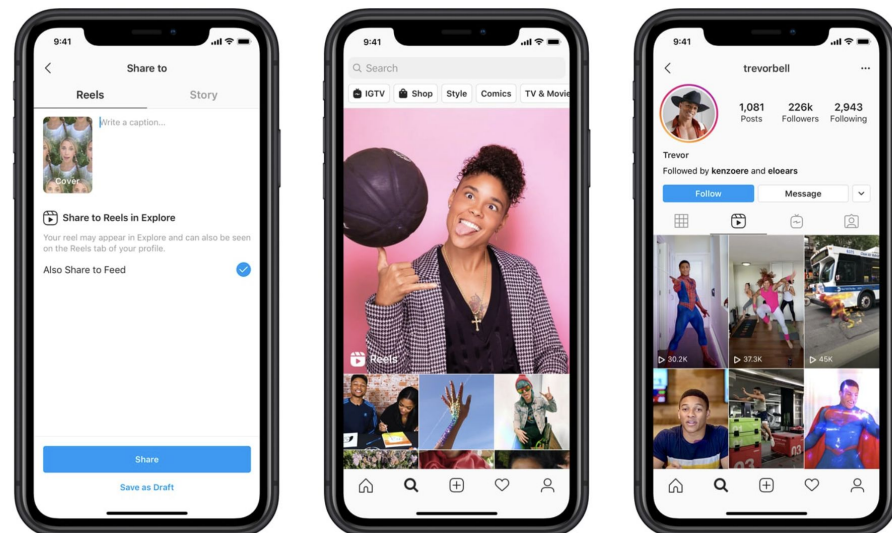
S. Jiang, Z. Shakeri, Y. Xia, et al, ResPROMPT: Residual Connection Prompting Advances Multi-Step Reasoning in Large Language Models, NAACL 2024



Description: *A group of toys embark on a journey of self-discovery as they learn the true meaning of friendship.*

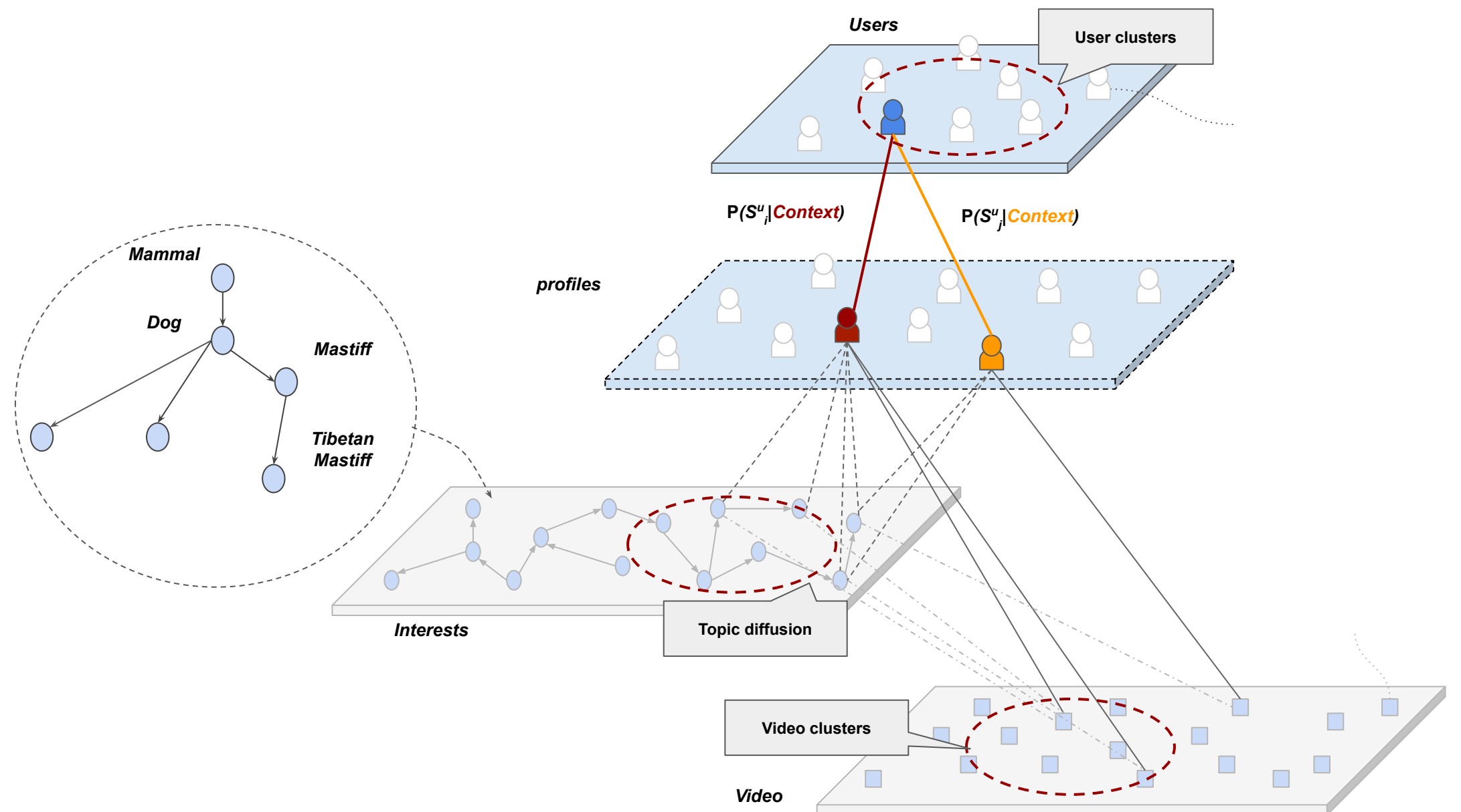


Case study: Graph for user interest modeling



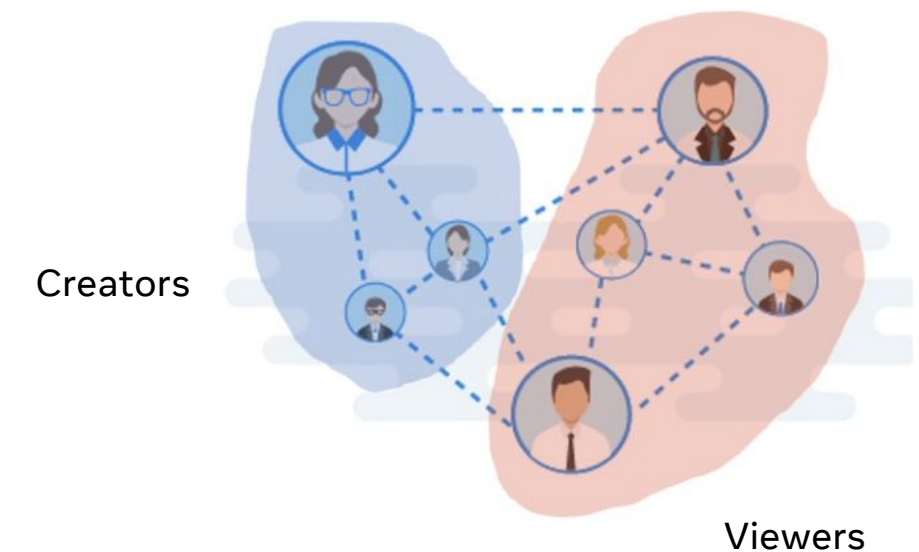
<https://about.instagram.com/blog/announcements/introducing-instagram-reels-announcement>

Graphs provide advanced modeling and system techniques for holistic user interest understanding and personalized recommendation, which provides interpretability to the output of some SOTA ML models, as well as supporting better user control over recommendations.



Case study: Graph ranking for audience matching

- Graph representation
 - Creator, viewer, media \Rightarrow heterogeneous nodes
 - Following, engagement, profile matching \Rightarrow edges
- Graph ranking
 - For a given creator (or a creator's newly shared media), rank the viewers within the neighborhood of the creator
 - Heterogeneous PPR with the creator as a root
 - GCN message passing with rich feature support



Case study:

Graph ranking for audience matching (2)

- Personalized PageRank (PPR) helps find related creators for a given viewer (source node) in audience matching

■ $r_s = (1 - c) * e_s + c * A * r_s$

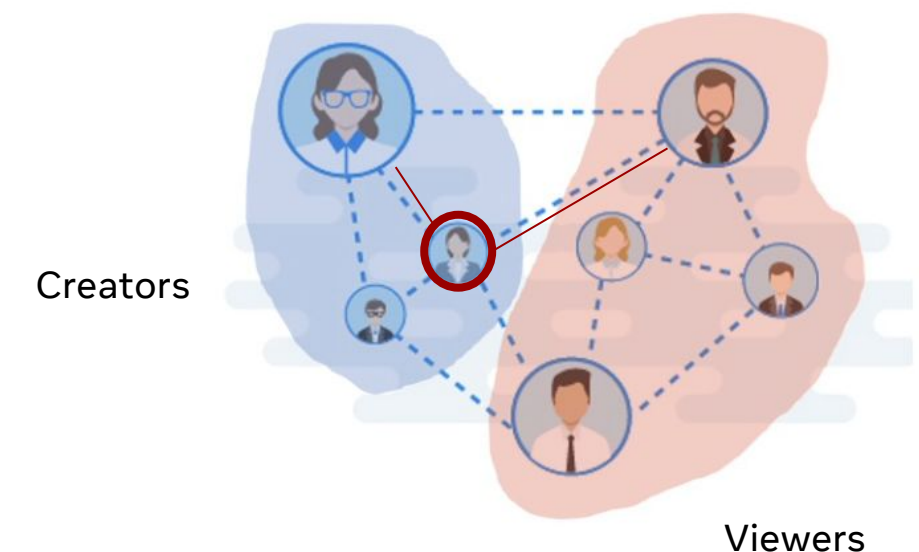
⇒ $ppr_s(creator) = (1 - c) * e_s + c * \sum_{creator' \in path(creator-v-creator')} ppr_s(creator') / deg(creator')$

- A: the adjacency matrix
- s: the source node, say, a given viewer
- c: damping factor, e.g. 0.85
- e_s : biased teleportation vector, e.g. $[0, 0, \dots, 1, \dots, 0]$ where $e_s[s]=1$
- r_s : vector of probability showing the relevance of v to source, i.e., $ppr[v]$
- $deg(creator')$: #path

■ What is missing?

- $deg(u)$ reflects interest match, not the size of connected followers
- $deg(u)$ does not reflect the “size/status” of a creator i.e. new /casual/power creator

■ Treatment?

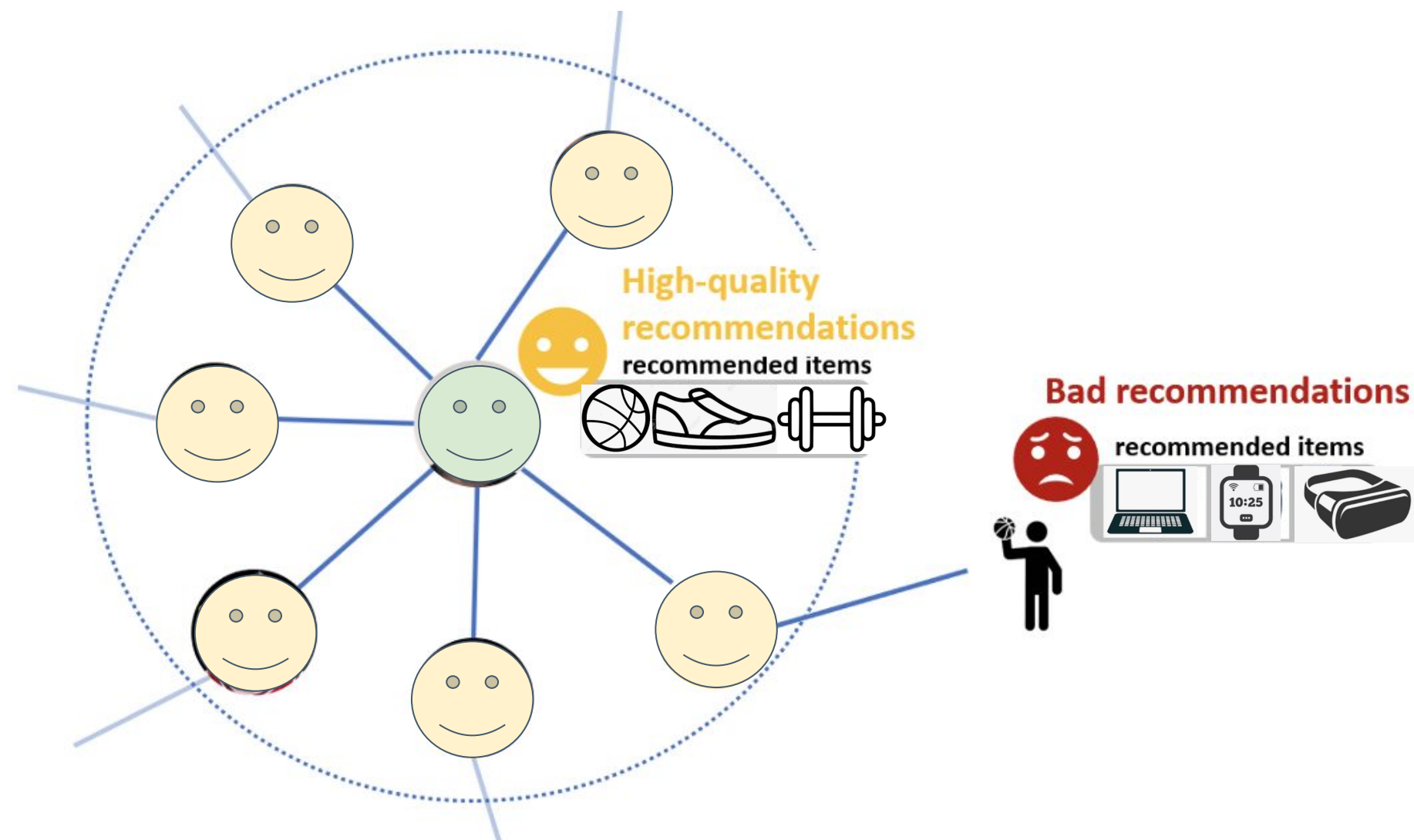


Agenda

- 01 Graph in RecSys
- 02 **Graph sparse nodes in recommendation**
- 03 GNN expressiveness: depth & scope
- 04 Conclusion & Opportunities

Disparity of node degrees in GCN

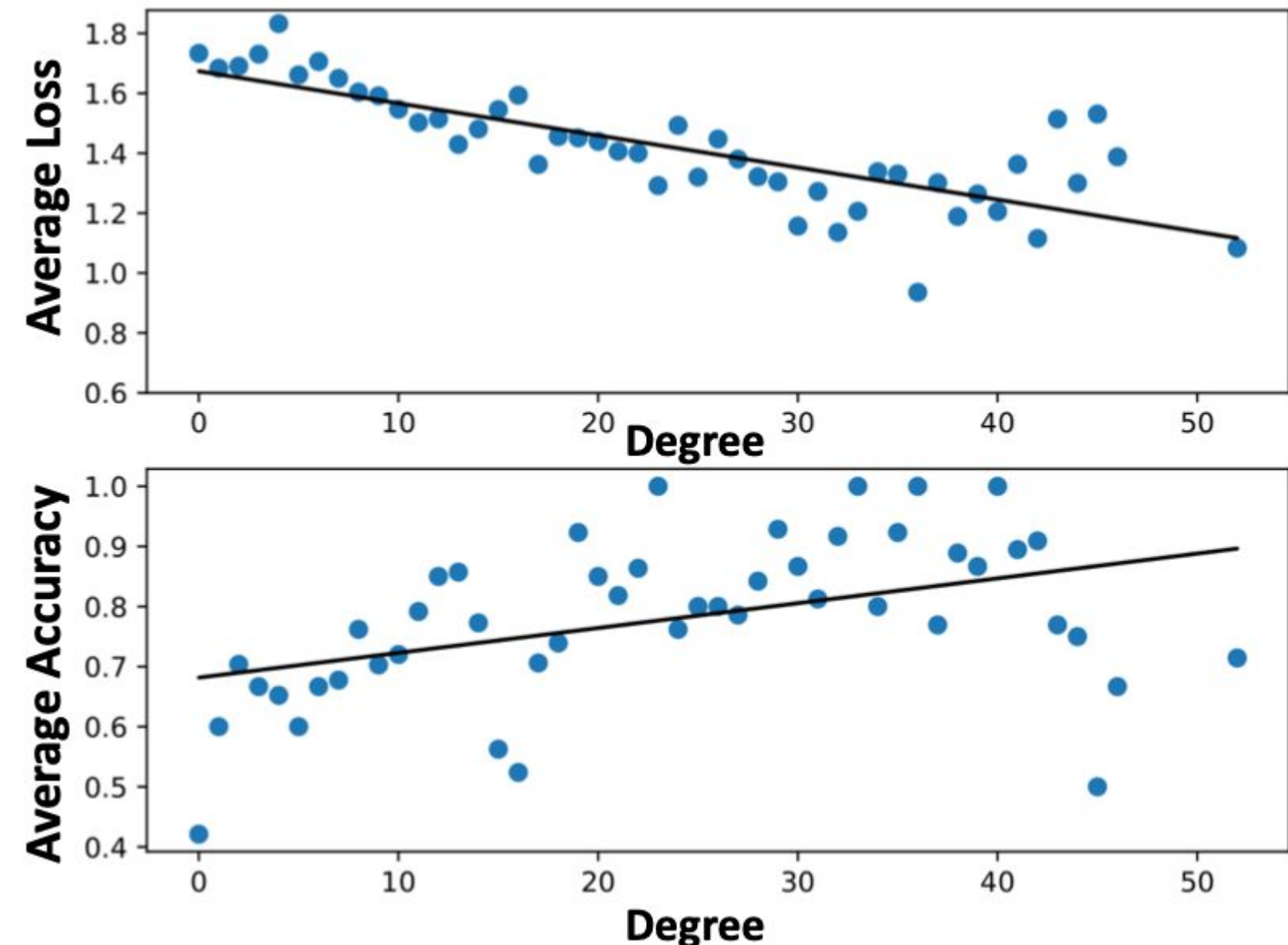
- Another example: Online advertising
 - Celebrities often enjoy high-quality recommendations
 - Grassroot users often suffer from bad recommendations



Disparity of node degrees in GCN

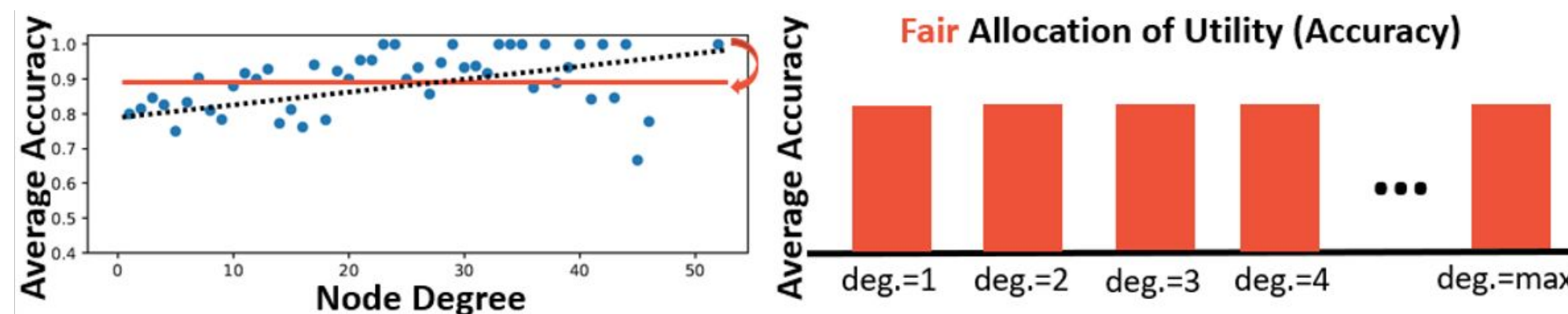
- Illustrative example
 - 2-layer GCN on Amazon-Photo
 - Average loss/accuracy of degree groups
- Observation
 - Low-degree nodes suffer from higher loss and lower accuracy
 - Real-world graph is often long-tailed
 - GCN might be primarily beneficial to a few high-degree nodes

If small-sized creators are what we are interested in, then the GCN framework needs some remedy.



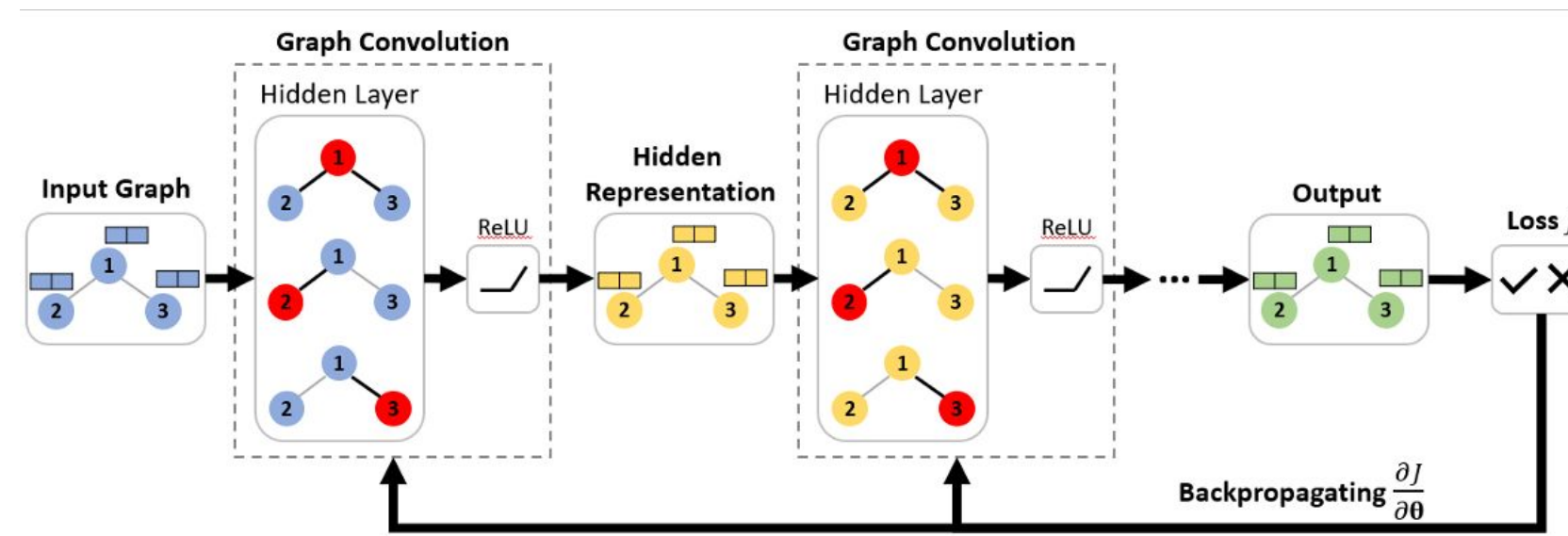
Rawlsian difference principle for GCN

- Given
 - An undirected graph $\mathcal{G}=(\mathbf{A},\mathbf{X})$
 - An L -layer GCN with weights θ
 - A task-specific loss J
- Find: A well-trained GCN that
 - Minimizes the task-specific loss
 - Achieves a fair allocation of utility for the groups of nodes with the same degree
- Key question:
 - When is the allocation of utility fair?



Understand the loss

- **Intuition:** Understand why the loss varies **after training**
- **What happens during training?**
 - Extract node representations
 - Predict the outcomes using the node representations
 - Calculate the task-specific loss J
 - Update model weights θ by the gradient $\partial J / \partial \theta \leftarrow$ key component for training
- **Question:** Is the unfairness caused by the gradient?

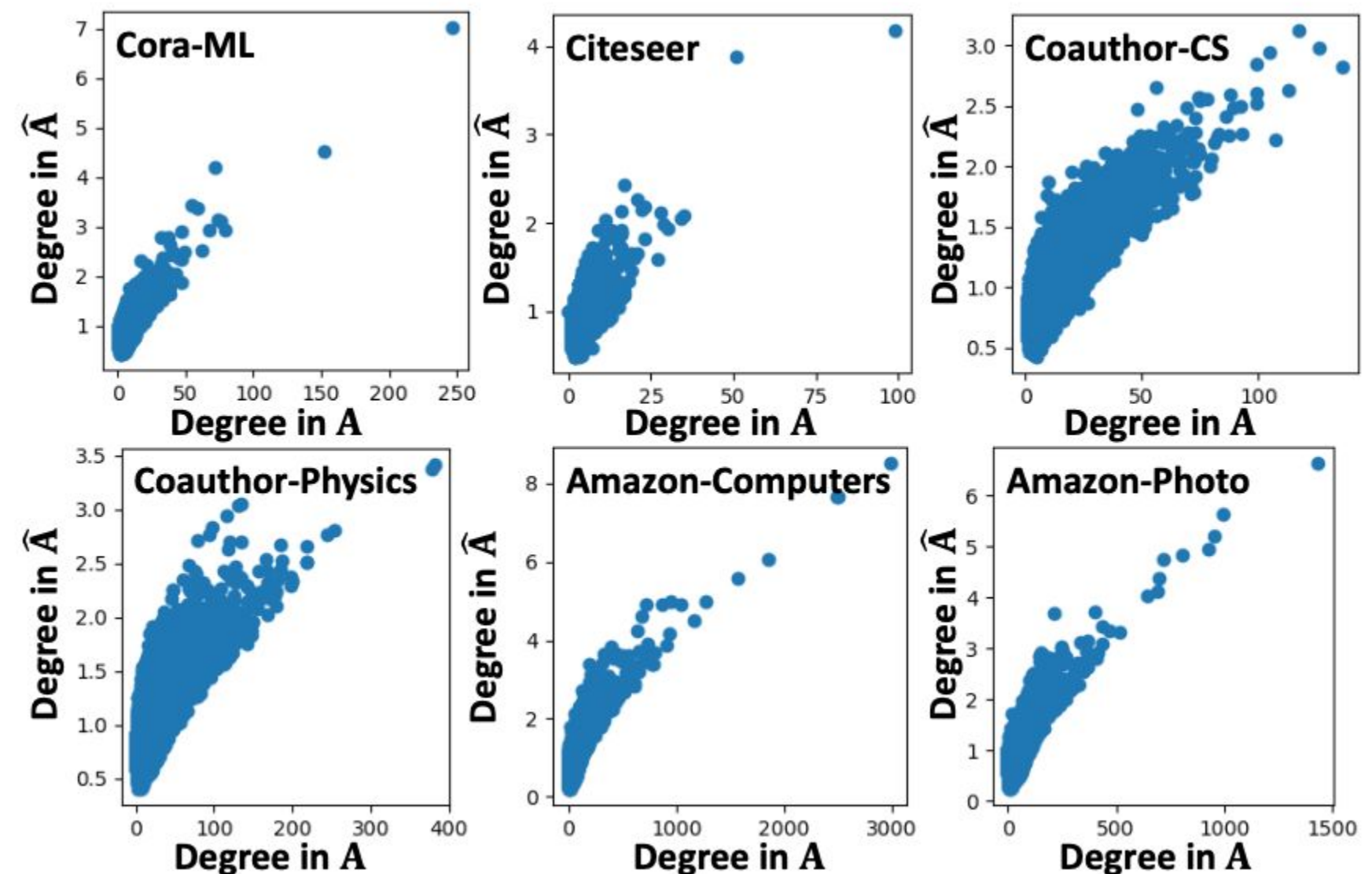


Rawlsian difference principle for GCN

- Rawlsian difference principle
 - Equality principle in distributive justice
 - Equality \Rightarrow maximize welfare of worst-off
- $$\text{grc } \min_{\theta} \text{Var}(\{U(\mathcal{D}_i, \theta) | i = 1, \dots, h\})$$
- $$\text{s.t. } \theta = \text{argmin } J(\mathcal{D}, Y, \theta)$$
- Importance on the gradient
 - $\nabla \frac{\partial J}{\partial \mathbf{W}^{(l)}} = \sum_{j=1}^n \text{deg}_{\hat{A}}(j) \mathbf{I}_j^{(\text{row})} = \sum_{i=1}^n \text{deg}_{\hat{A}}(i) \mathbf{I}_i^{(\text{col})}$ on nodes

■ \mathbf{I}_j is a degree-free influence of node j

Higher node degree in A , more importance it has on the gradient of the weight matrix.



Rawlsian GCN

- Doubly Stochastic Matrix Computation Normalize the importance of influence matrices to mitigate the node degree-related unfairness, so that each node will have equal importance in updating the weight parameters
 - Computing the doubly stochastic matrix is nontrivial, so we adopt the Sinkhorn-Knopp
- Rawlsian GCN
 - Gradient is computed using the doubly stochastic matrix, so that all nodes will have equal importance in determining the gradient

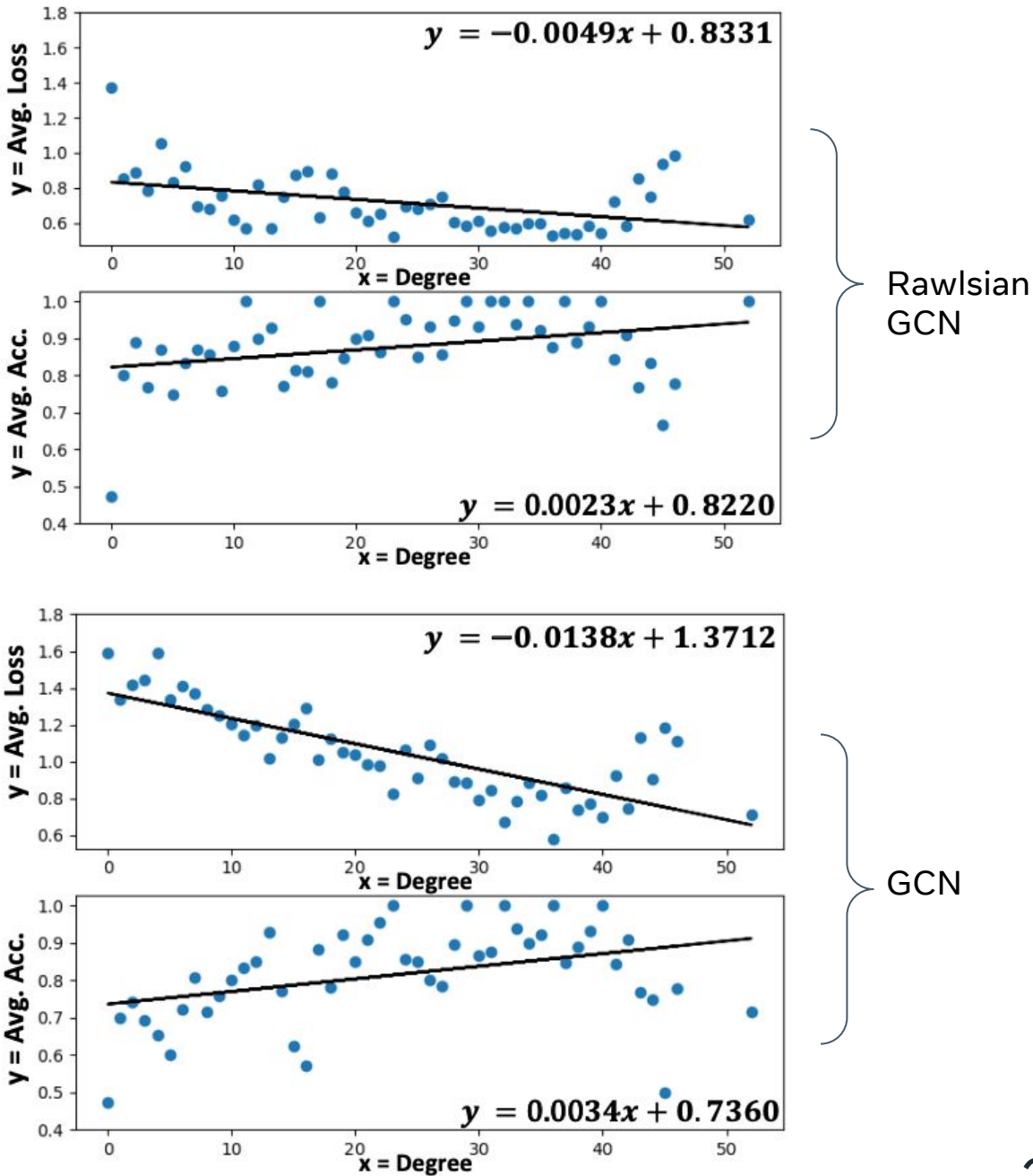
$$\frac{\partial J}{\partial \mathbf{W}^{(l)}_{\text{fair}}} = (\mathbf{H}^{(l-1)})^T \hat{\mathbf{A}}_{DS}^T \frac{\partial J}{\partial \mathbf{E}^{(l)}}$$

$$\text{where } \mathbf{E}^{(l)} = \hat{\mathbf{A}} \mathbf{H}^{(l-1)} \mathbf{W}^{(l)}.$$

Experiments

| Method | Cora-ML | | Citeseer | | Coauthor-CS | |
|-----------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | Acc. | Bias | Acc. | Bias | Acc. | Bias |
| GCN | 80.10 ± 0.812 | 0.392 ± 0.046 | 68.60 ± 0.341 | 0.353 ± 0.040 | 93.28 ± 0.194 | 0.075 ± 0.004 |
| DEMO-Net | 61.60 ± 0.687 | 0.181 ± 0.015 | 60.26 ± 0.408 | 0.315 ± 0.022 | 65.90 ± 0.583 | 0.164 ± 0.006 |
| DSGCN | 30.26 ± 5.690 | 8.003 ± 2.766 | 31.42 ± 3.257 | 6.887 ± 1.947 | 44.20 ± 7.155 | 1.460 ± 0.397 |
| Tail-GNN | 78.54 ± 0.582 | 0.503 ± 0.284 | 66.34 ± 0.009 | 0.655 ± 0.382 | 92.66 ± 0.196 | 0.052 ± 0.031 |
| AdvFair | 67.56 ± 2.594 | 10.01 ± 2.480 | 50.26 ± 6.277 | 3.146 ± 2.425 | 84.82 ± 2.254 | 12.26 ± 6.797 |
| REDRESS | 75.70 ± 0.620 | 0.955 ± 0.213 | 65.80 ± 0.518 | 0.944 ± 0.077 | 92.44 ± 0.233 | 0.028 ± 0.003 |
| RAWLsGCN-Graph (Ours) | 76.96 ± 1.098 | 0.105 ± 0.012 | 69.34 ± 0.745 | 0.196 ± 0.013 | 92.52 ± 0.264 | 0.043 ± 0.002 |
| RAWLsGCN-Grad (Ours) | 79.34 ± 1.247 | 0.232 ± 0.065 | 68.81 ± 0.462 | 0.283 ± 0.047 | 92.68 ± 0.240 | 0.058 ± 0.007 |

| Method | Coauthor-Physics | | Amazon-Computers | | Amazon-Photo | |
|-----------------------|------------------|---------------|------------------|---------------|---------------|---------------|
| | Acc. | Bias | Acc. | Bias | Acc. | Bias |
| GCN | 93.96 ± 0.367 | 0.023 ± 0.001 | 64.84 ± 0.641 | 0.353 ± 0.026 | 79.58 ± 1.507 | 0.646 ± 0.038 |
| DEMO-Net | 77.50 ± 0.566 | 0.084 ± 0.010 | 26.48 ± 3.455 | 0.456 ± 0.021 | 39.92 ± 1.242 | 0.243 ± 0.013 |
| DSGCN | 79.08 ± 1.533 | 0.262 ± 0.075 | 27.68 ± 1.663 | 1.407 ± 0.685 | 26.76 ± 3.387 | 0.921 ± 0.805 |
| Tail-GNN | OOM | OOM | 76.24 ± 1.491 | 1.547 ± 0.670 | 86.00 ± 2.715 | 0.471 ± 0.264 |
| AdvFair | 87.44 ± 1.132 | 0.892 ± 0.502 | 53.50 ± 5.362 | 4.395 ± 1.102 | 75.80 ± 3.563 | 51.24 ± 39.94 |
| REDRESS | 94.48 ± 0.172 | 0.019 ± 0.001 | 80.36 ± 0.206 | 0.455 ± 0.032 | 89.00 ± 0.369 | 0.186 ± 0.030 |
| RAWLsGCN-Graph (Ours) | 94.06 ± 0.196 | 0.016 ± 0.000 | 80.16 ± 0.859 | 0.121 ± 0.010 | 88.58 ± 1.116 | 0.071 ± 0.006 |
| RAWLsGCN-Grad (Ours) | 94.18 ± 0.306 | 0.021 ± 0.002 | 74.18 ± 2.530 | 0.195 ± 0.029 | 83.70 ± 0.672 | 0.186 ± 0.068 |

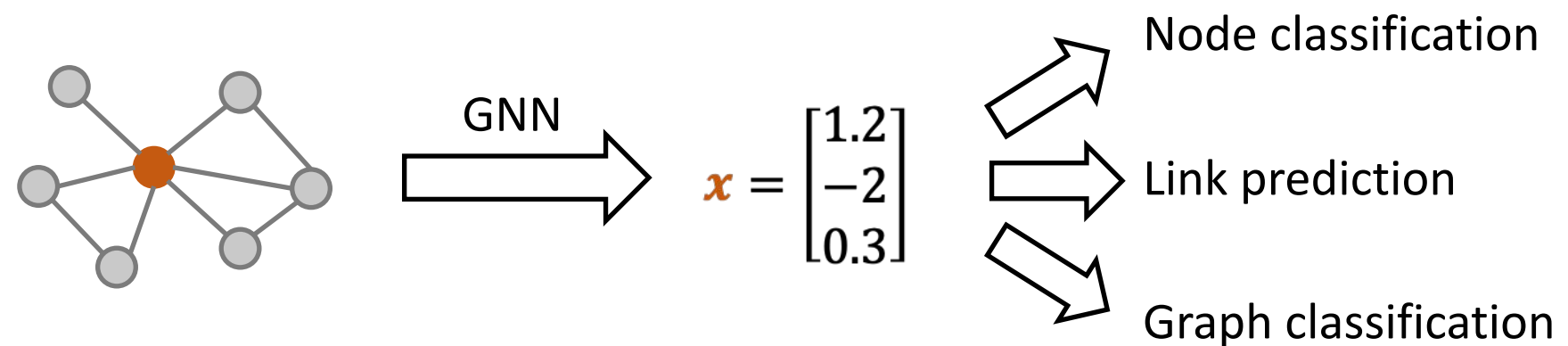


Agenda

- 01 Graph in RecSys
- 02 Graph sparse nodes in recommendation
- 03 **GNN expressiveness: depth & scope**
- 04 Conclusion & Opportunities

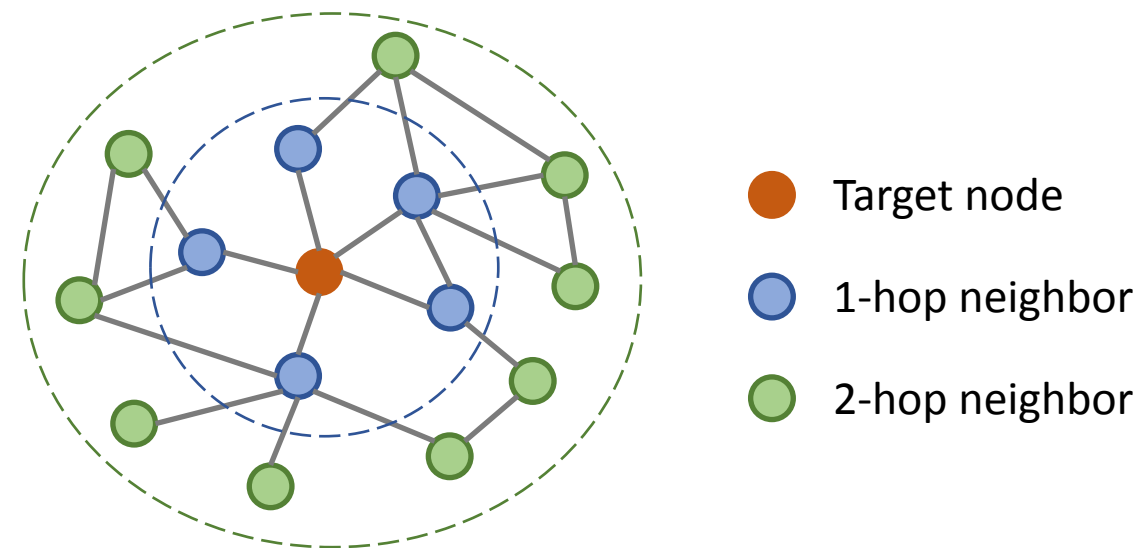
Recap: Graph Neural Networks

Graph representation learning



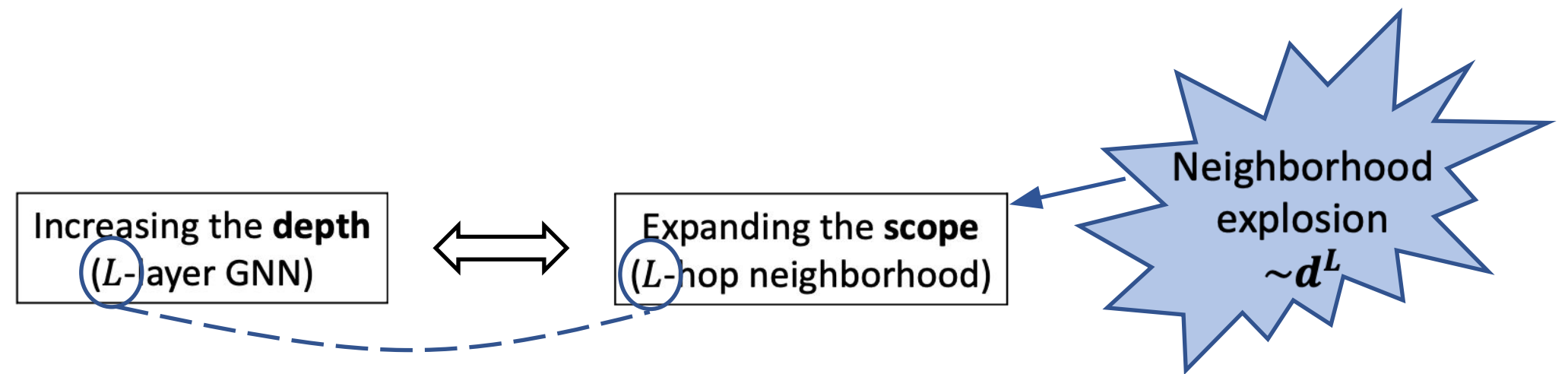
Message passing in GNNs

- **Scope:** from what neighbors?
- **Depth:** by how many iterations / layers?



Scalability & Expressivity Challenges

GNN designs by default (on large scale graphs):



Dilemma in deep GNN: scalability-expressivity tradeoff

- **Depth is important:** Experience from general deep learning
- **Depth is expensive:** Observation from graph message passing
- **Depth can cause training challenges:** Oversmoothing in GCN

Solution: Don't forget the scope!

Depth-Scope Decoupling

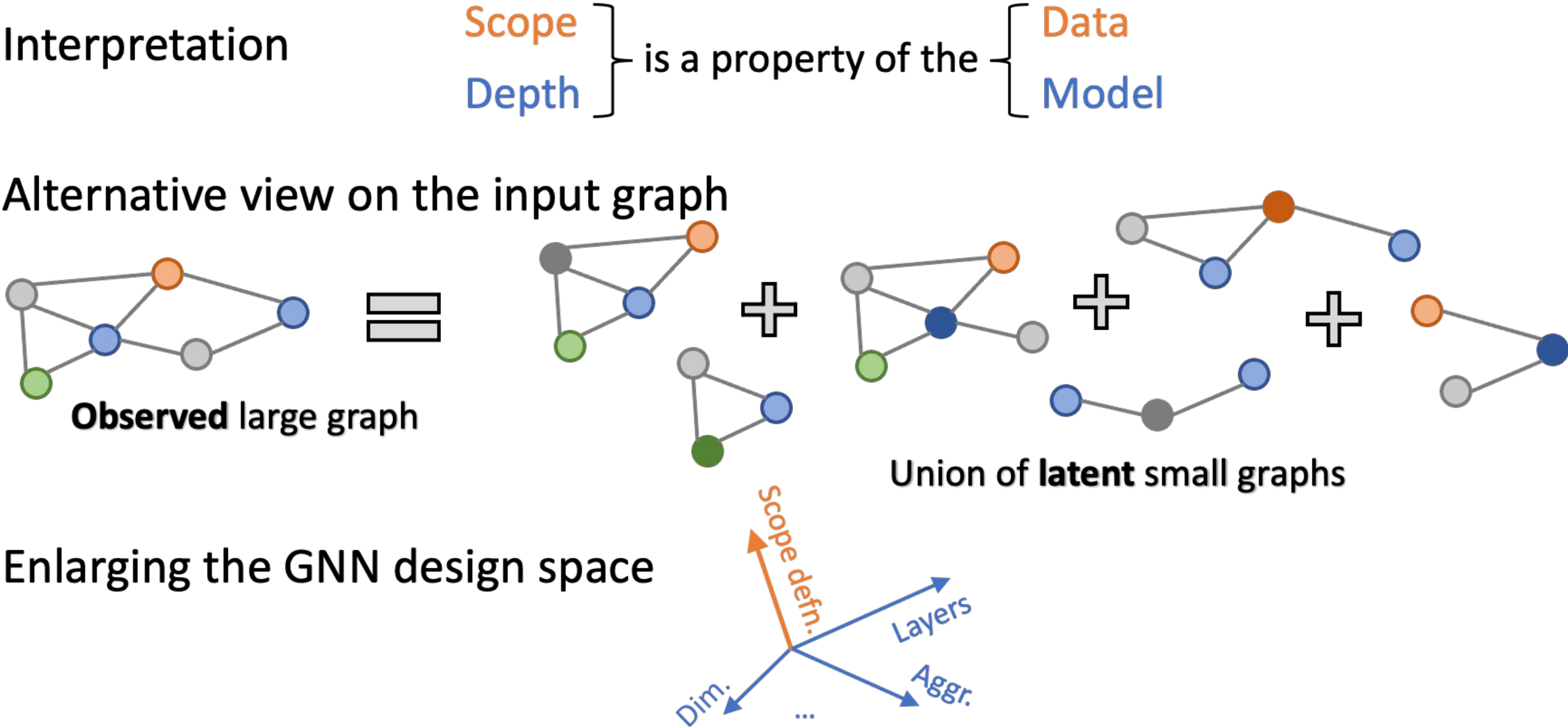
Define ***scope*** independent of ***depth***

- Intuitions
 - Some neighbors are irrelevant \rightarrow no need to pass their messages
 - Some neighbors are extra important \rightarrow worth passing their messages many times
- Example: Deep (L' -layer) GNN on shallow (L -hop) subgraph

Algorithm: generate embedding for a target node v of the full graph \mathcal{G}

1. Extract a subgraph $\mathcal{G}_{[v]}$ around v
2. for round $i = 1$ to L' :
 Perform message passing along all edges in $\mathcal{G}_{[v]}$
3. Take v 's embedding from all node embeddings of $\mathcal{G}_{[v]}$

Depth-Scope Decoupling



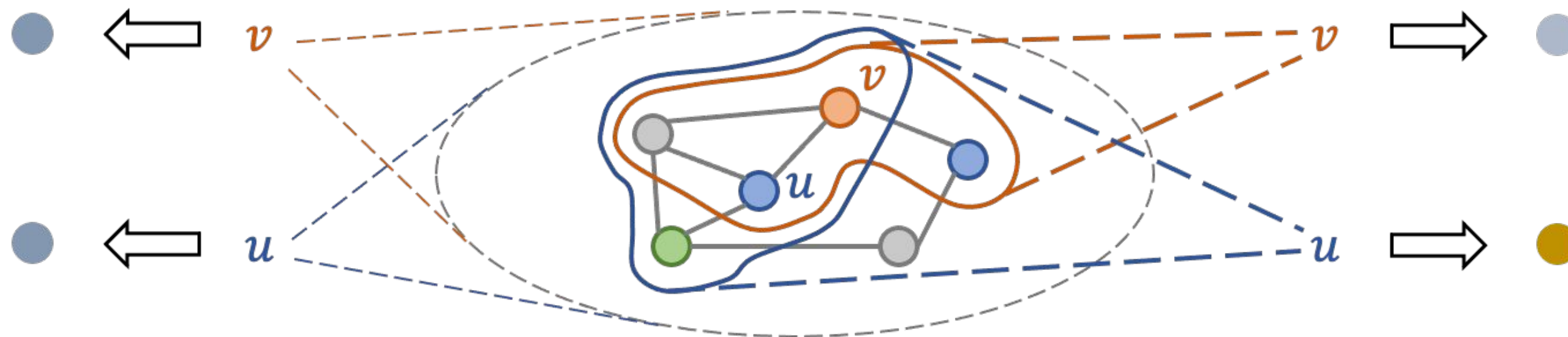
Theoretical Justification: Graph Signal Processing Perspective

Oversmoothing of GCN

- Each GCN layer smooths features of direct neighbors
- For $u \neq v$, their scopes are both the full graph G
- Many GCN layers smooths features of the full graph G

Non-smoothing of decoupled-GCN

- GCN layers only smooths the features within $G_{[v]}$
- For $u \neq v$, their scopes can be different $G_{[u]} \neq G_{[v]}$
- Smoothing different set of features produces distinctive embeddings



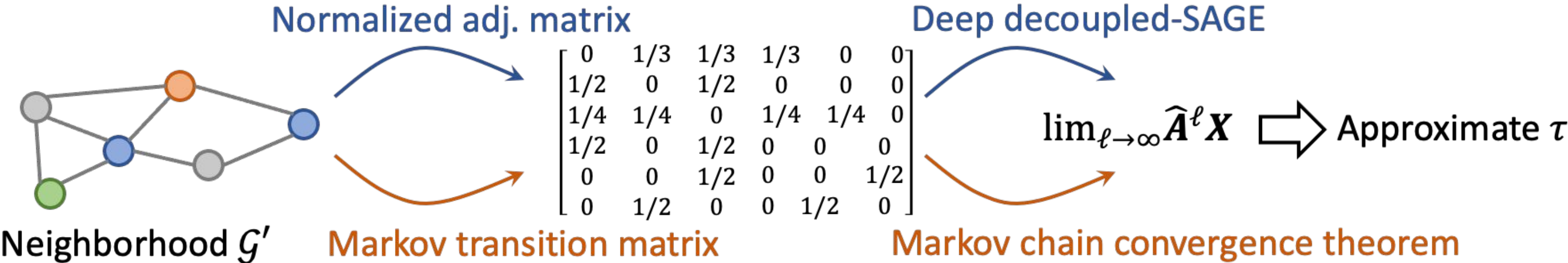
Theoretical Justification: Function Approximator Perspective

Decoupled-SAGE is more expressive than GraphSAGE

- Consider neighborhood G' & function τ for linear comb. of G' features
- GraphSAGE cannot approximate τ well, even if G' is L -hop neighborhood
- Decoupled-SAGE can approximate τ where

Scope $\mathcal{G}_{[v]} = \mathcal{G}'$

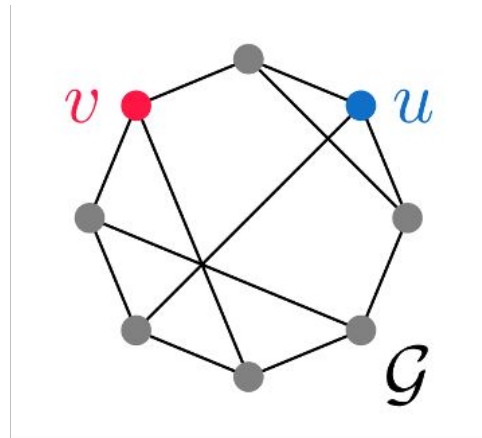
Depth reduces the error exponentially



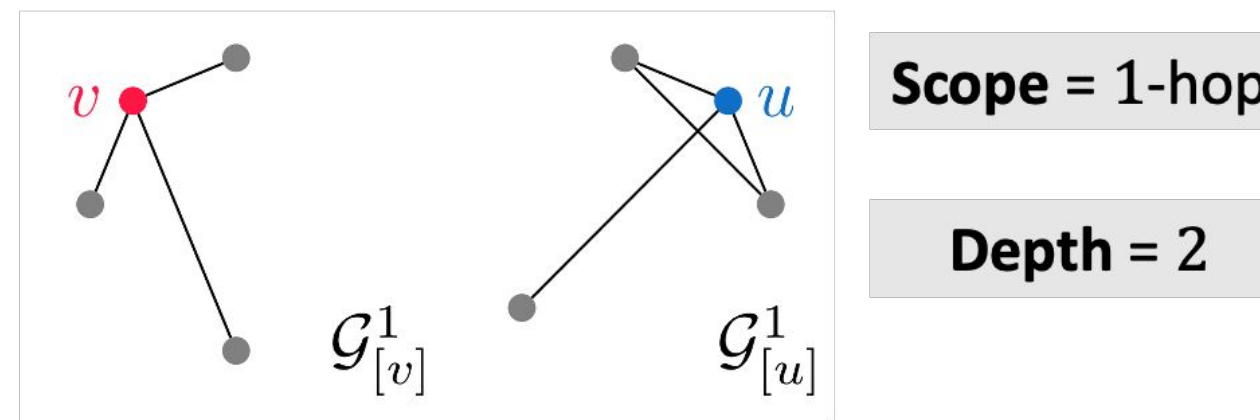
Theoretical Justification: Topology Information Perspective

Decoupled-GIN is more expressive than GIN/1-WL

- Challenge for GIN/1-WL: non-isomorphic regular graphs
- Benefit of decoupling: *subgraphs of a regular graph may not be regular*



Example 3-regular graph where GIN cannot distinguish u and v



Decoupled-GIN can distinguish u and v

Evaluation: Setup

Datasets 7 graphs (up to 111M nodes)

inductive & transductive

Backbone models 5 aggregation functions & residue connection

Tasks node classification & link prediction

Training of baselines full batch & GraphSAINT minibatch

Training of proposed minibatch of independently constructed $G_{[v]}$

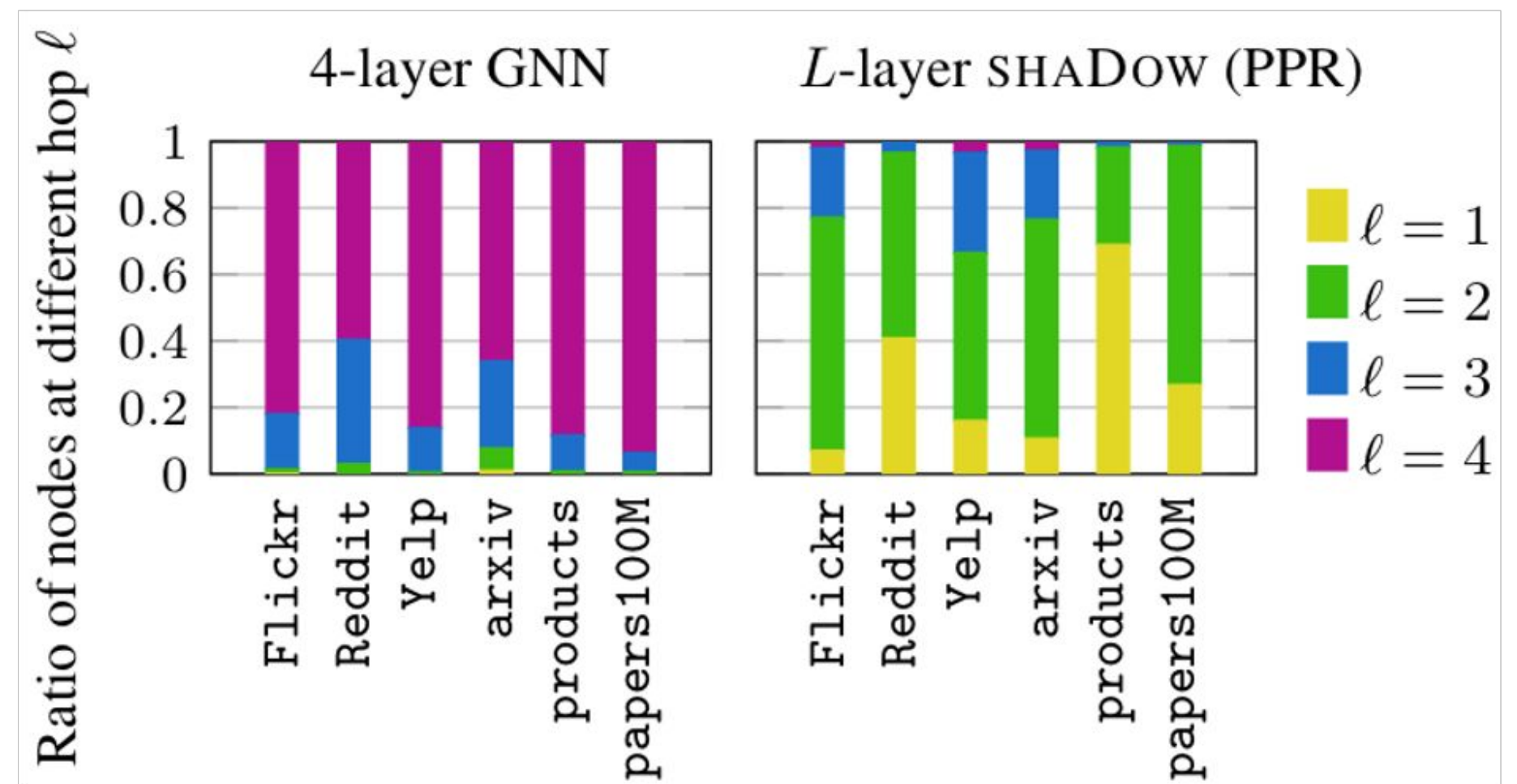
Practical design: shaDow-GNN (Decoupled GNN on shallow subgraphs)

- Scope: based on 2-hop / PPR (top 200 nodes)
- Depth: 3- / 5-layer

Evaluation: Neighborhood Composition

How many neighbors are ℓ hops away from the target node?

- Scope of normal GNN
 - Dominated by distant neighbors
 - Size grows rapidly
- Scope of shaDow-GNN
 - Consists of nearby neighbors
 - Size is small regardless of number of layers (< 200 neighbors)



Evaluation: Baseline Comparisons

- Decoupling improves **accuracy** at lower computation **cost**
- Decoupling is a general design principle applicable to **various backbones**
- **Subgraph extraction** algorithms are important

| Method | Layers | Flickr | | Reddit | | Yelp | | ogbn-arxiv | | ogbn-products | |
|------------------------------|--------|-----------------------|------------|-----------------------|------------|-----------------------|------------|-----------------------|------------|-----------------------|------------|
| | | Accuracy | Cost | Accuracy | Cost | F1-micro | Cost | Accuracy | Cost | Accuracy | Cost |
| GCN | 3 | 0.5159±0.0017 | 2E0 | 0.9532±0.0003 | 6E1 | 0.4028±0.0019 | 2E1 | 0.7170±0.0026 | 1E1 | 0.7567±0.0018 | 5E0 |
| | 5 | 0.5217±0.0016 | 2E2 | 0.9495±0.0012 | 1E3 | OOM | 1E3 | 0.7186±0.0017 | 1E3 | OOM | 9E2 |
| GCN + GraphSAINT-RW | 3 | 0.5155±0.0027 | 2E0 | 0.9523±0.0003 | 6E1 | 0.5110±0.0012 | 2E1 | 0.7093±0.0003 | 1E1 | 0.8003 ±0.0024 | 5E0 |
| | 5 | 0.5165±0.0026 | 2E2 | 0.9523±0.0012 | 1E3 | 0.5012±0.0021 | 1E3 | 0.7039±0.0020 | 1E3 | 0.7992±0.0021 | 9E2 |
| SHADow-GCN + PPR | 3 | 0.5262±0.0018 | (1) | 0.9581±0.0004 | (1) | 0.5255±0.0012 | (1) | 0.7192±0.0025 | (1) | 0.7778±0.0030 | (1) |
| | 5 | 0.5270 ±0.0024 | 1E0 | 0.9583 ±0.0002 | 1E0 | 0.5272 ±0.0018 | 2E0 | 0.7207 ±0.0030 | 2E0 | 0.7844±0.0029 | 2E0 |
| GraphSAGE | 3 | 0.5140±0.0014 | 3E0 | 0.9653±0.0002 | 5E1 | 0.6178±0.0033 | 2E1 | 0.7192±0.0027 | 1E1 | 0.7858±0.0013 | 4E0 |
| | 5 | 0.5154±0.0052 | 2E2 | 0.9626±0.0004 | 1E3 | OOM | 2E3 | 0.7193±0.0037 | 1E3 | OOM | 1E3 |
| GraphSAGE + GraphSAINT-RW | 3 | 0.5176±0.0032 | 3E0 | 0.9671±0.0003 | 5E1 | 0.6453±0.0011 | 2E1 | 0.7107±0.0003 | 1E1 | 0.7923±0.0023 | 4E0 |
| | 5 | 0.5201±0.0032 | 2E2 | 0.9670±0.0010 | 1E3 | 0.6394±0.0003 | 2E3 | 0.7013±0.0021 | 1E3 | 0.7964±0.0022 | 1E3 |
| SHADow-SAGE + 2-hop | 3 | 0.5288±0.0014 | 1E0 | 0.9660±0.0003 | 1E0 | 0.6493±0.0001 | 1E0 | 0.7163±0.0012 | 1E0 | 0.7993±0.0012 | 1E0 |
| | 5 | 0.5338±0.0038 | 2E0 | 0.9661±0.0002 | 2E0 | 0.6503±0.0001 | 2E0 | 0.7183±0.0012 | 2E0 | 0.8014±0.0020 | 2E0 |
| SHADow-SAGE + PPR | 3 | 0.5344±0.0028 | (1) | 0.9693 ±0.0002 | (1) | 0.6512 ±0.0002 | (1) | 0.7234±0.0032 | (1) | 0.7945±0.0021 | (1) |
| | 5 | 0.5424 ±0.0025 | 2E0 | 0.9691±0.0003 | 2E0 | 0.6502±0.0001 | 2E0 | 0.7255 ±0.0013 | 2E0 | 0.8043 ±0.0026 | 2E0 |
| GAT | 3 | 0.5070±0.0032 | 2E1 | OOM | 3E2 | OOM | 2E2 | 0.7201±0.0011 | 1E2 | OOM | 3E1 |
| | 5 | 0.5164±0.0033 | 2E2 | OOM | 2E3 | OOM | 2E3 | OOM | 3E3 | OOM | 2E3 |
| GAT + GraphSAINT-RW | 3 | 0.5225±0.0053 | 2E1 | 0.9671±0.0003 | 3E2 | 0.6459±0.0002 | 2E2 | 0.6977±0.0003 | 1E2 | 0.8027±0.0028 | 3E1 |
| | 5 | 0.5153±0.0034 | 2E2 | 0.9651±0.0024 | 2E3 | 0.6478±0.0012 | 2E3 | 0.6954±0.0013 | 3E3 | 0.7990±0.0072 | 2E3 |
| SHADow-GAT + PPR | 3 | 0.5383 ±0.0032 | (1) | 0.9703±0.0010 | (1) | 0.6549 ±0.0002 | (1) | 0.7243±0.0011 | (1) | 0.8014±0.0012 | (1) |
| | 5 | 0.5342±0.0023 | 2E0 | 0.9710 ±0.0008 | 2E0 | 0.6537±0.0004 | 2E0 | 0.7283 ±0.0027 | 2E0 | 0.8094 ±0.0034 | 2E3 |

Evaluation: Scaling to 100M-Node Graph

OGB Leaderboard comparison

- Higher accuracy
- Orders of magnitude smaller neighborhood size

Memory consumption

- Lowest in both CPU and GPU
- Train & inference the 100M graph on a low-end server

Table 2: Leaderboard comparison on papers100M

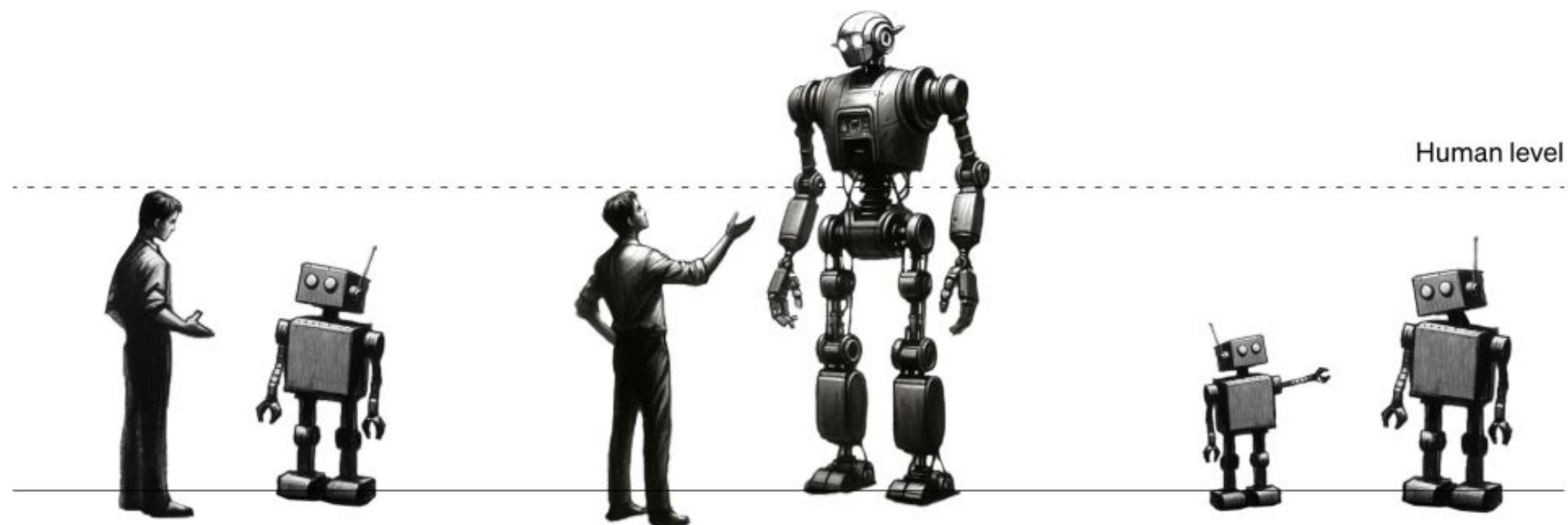
| Method | Test accuracy | Val accuracy | Neigh size |
|---------------------------|---------------------------------------|---------------------------------------|------------|
| GraphSAGE+incep | 0.6707 ± 0.0017 | 0.7032 ± 0.0011 | 4E5 |
| SIGN-XL | 0.6606 ± 0.0019 | 0.6984 ± 0.0006 | > 4E5 |
| SGC | 0.6329 ± 0.0019 | 0.6648 ± 0.0020 | > 4E5 |
| SHADOW-GAT ₂₀₀ | 0.6681 ± 0.0016 | 0.7019 ± 0.0011 | 2E2 |
| SHADOW-GAT ₄₀₀ | 0.6710 ± 0.0015 | 0.7067 ± 0.0012 | 3E2 |

Memory consumption of the ogbn-papers100M leaderboard methods

| Method | CPU RAM | GPU memory |
|-----------------|-------------|------------|
| GraphSAGE+incep | 80GB | 16GB |
| SIGN-XL | >682GB | 4GB |
| SGC | >137GB | 4GB |
| SHADOW-GAT | 80GB | 4GB |

Mixture of Weak and Strong on Graph

- Mixture-of-Experts (MoE) system has effectively improved model capacity
- Improve model capacity without significant trade-offs in computation overhead and optimization difficulty
- Achieve high expressive power, coming at the cost of minor computation overhead



Ref: OpenAI WEAK-TO-STRONG GENERALIZATION

3. Mixture of weak and strong

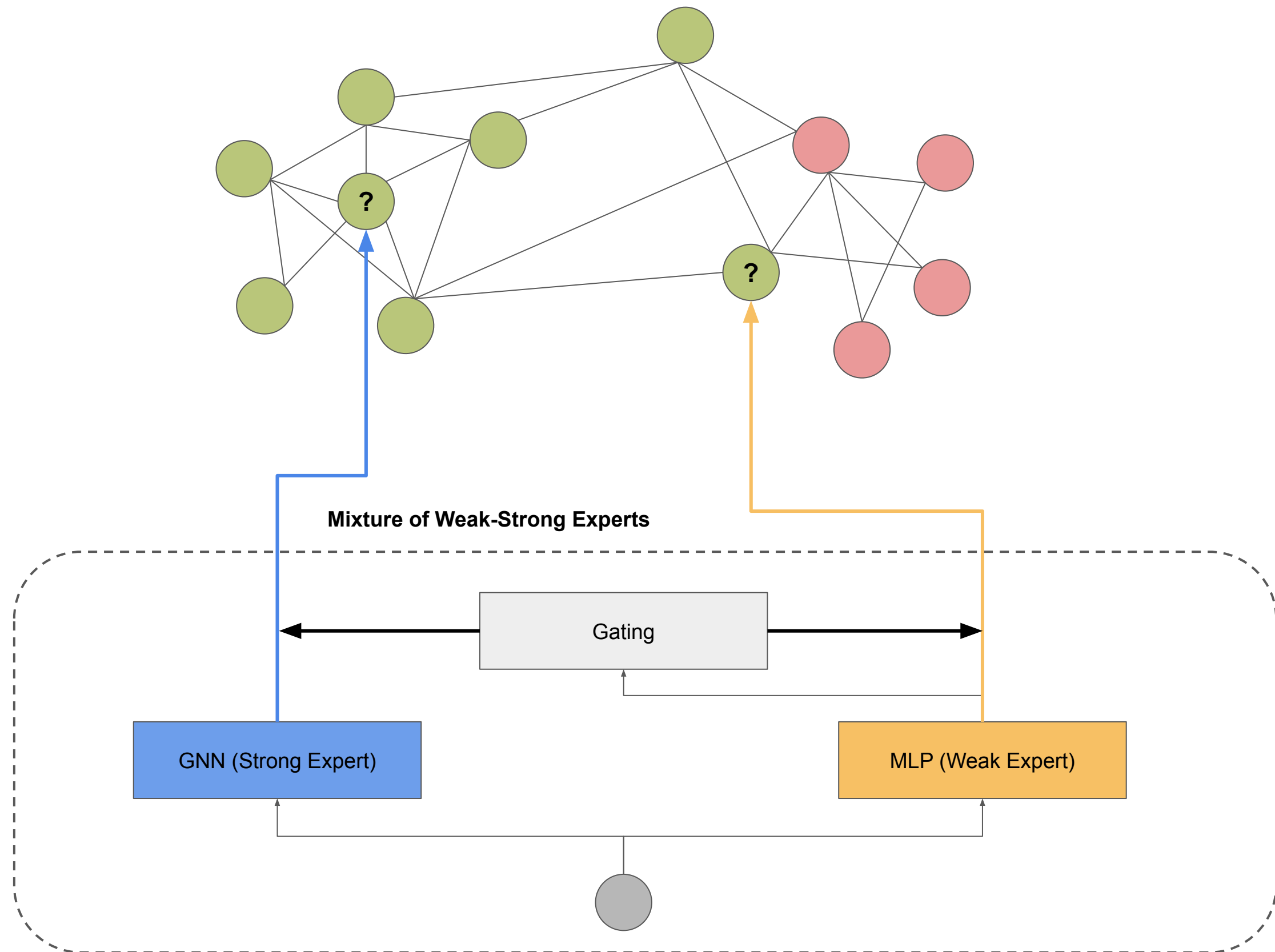
A novel mixture-of-experts design to combine a weak MLP expert with a strong GNN expert for decoupling the self-feature and neighbor structure modalities in graphs.

- Weak expert is a light-weight Multi-layer Perceptron (MLP)
- Strong expert is an off-the-shelf GNN

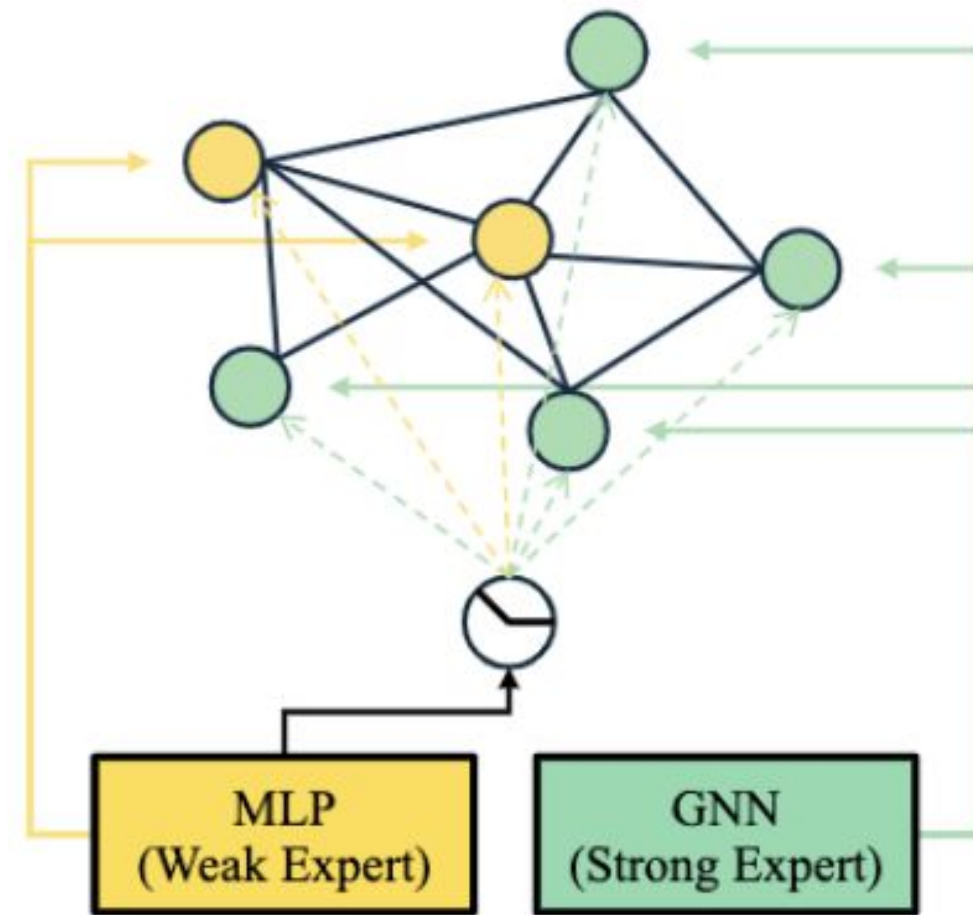
A “confidence” gating mechanism based on the dispersion of the weak expert’s prediction logits.

$$L = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} (C(\mathbf{p}_v) \cdot L(\mathbf{p}_v, \mathbf{y}_v) + (1 - C(\mathbf{p}_v)) \cdot L(\mathbf{p}'_v, \mathbf{y}_v))$$

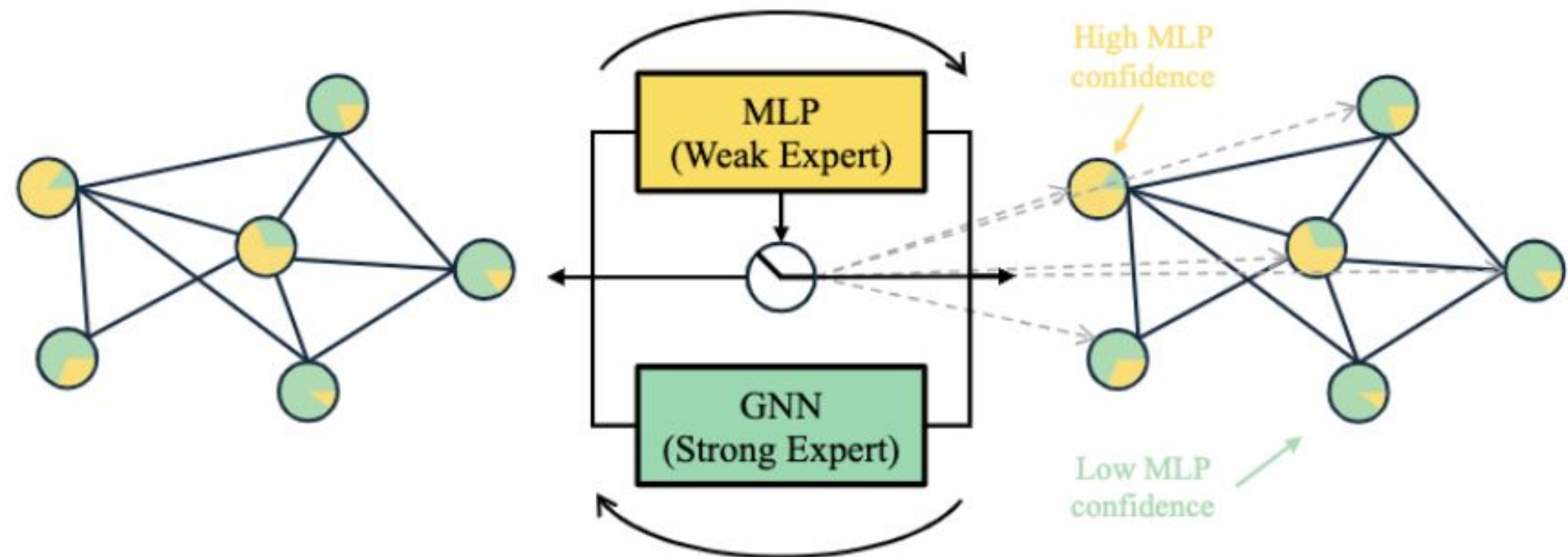
$$\mathbf{p}_v = \text{MLP}(\mathbf{x}_v; \boldsymbol{\theta}) \text{ and } \mathbf{p}'_v = \text{GNN}(\mathbf{x}_v; \boldsymbol{\theta}')$$



3. Mixture of weak and strong



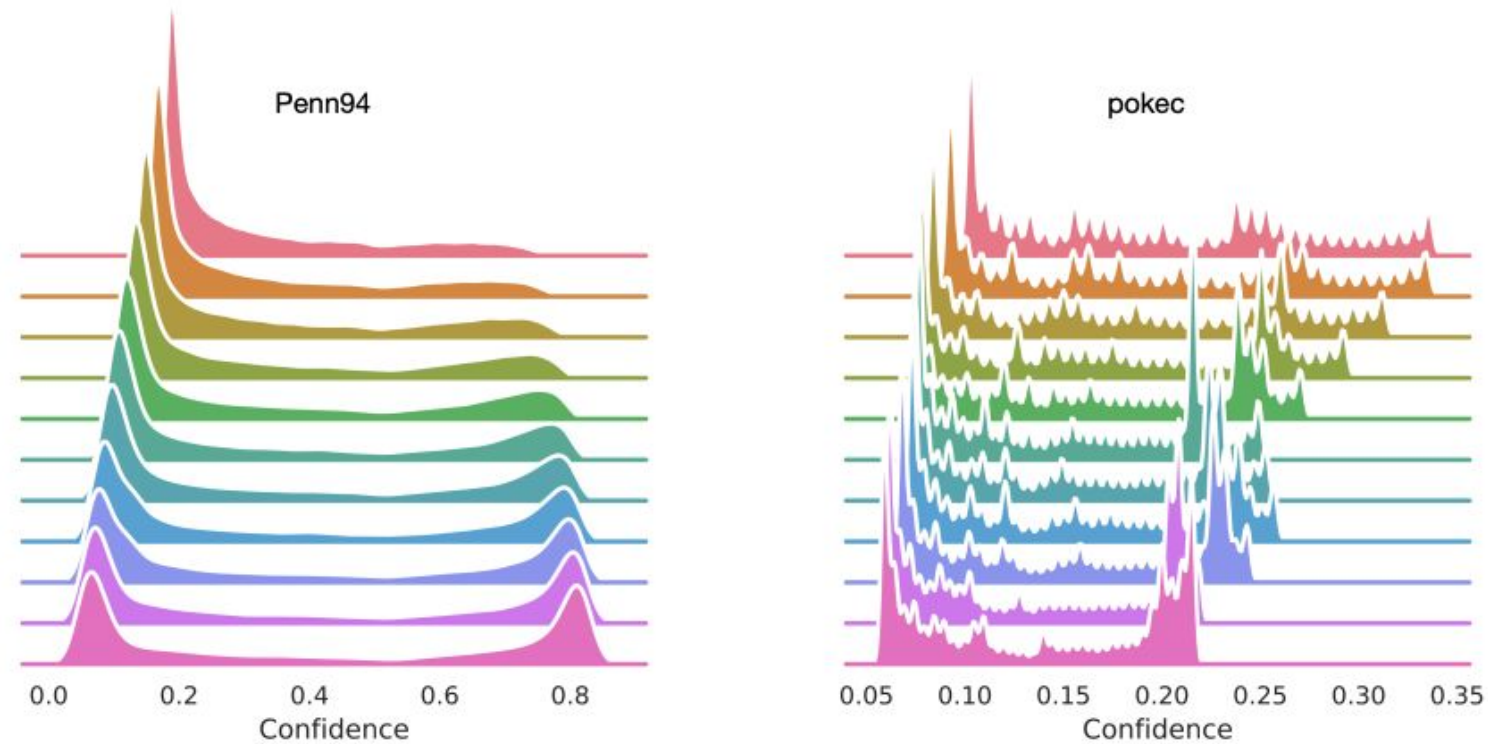
Fix GNN weights; Train MLP and Gating Module; Gating module computes MLP's confidence



Fix MLP and Gating module weights; Learn GNN; GNN focuses on nodes that MLP is *not* confident

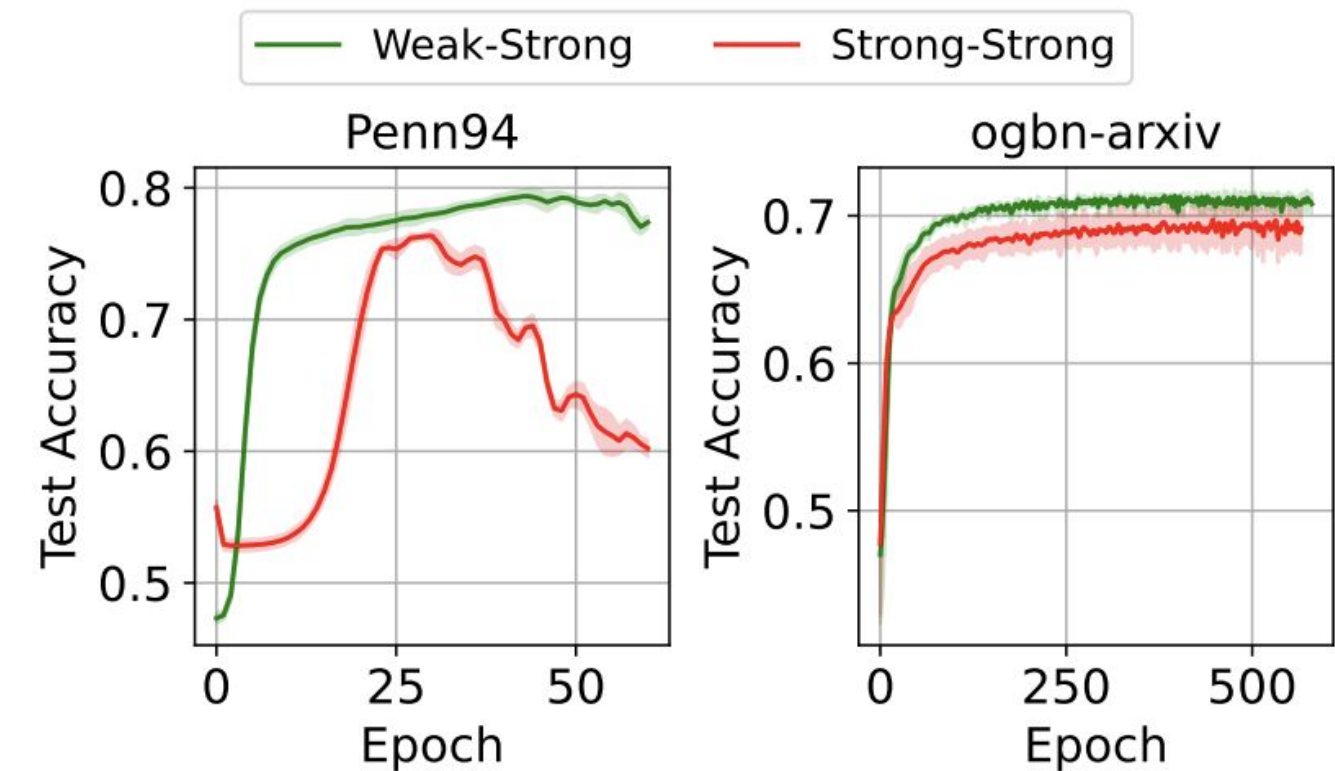
3. Mixture of weak and strong

Evaluation



On Penn94, the GNN dominates initially. The MLP gradually learns to specialize on a significant portion of the data. Eventually, the whole graph is clearly split between the two experts

For pokec, the specialization is not so strong. When training progresses, the MLP becomes less confidence when the GNN is optimized better; For different graphs, the two experts will adapt their extent of collaboration by minimizing the confidence-weighted loss.



On both Penn94 and ogbn-arxiv, we observe that the convergence quality of the “strong-strong” variant is much worse. The per-epoch execution time for “strong-strong” is also significantly longer, indicating an even longer overall convergence time than the “weak-strong” model.

Agenda

- 01 Graph in RecSys
- 02 Graph sparse nodes in recommendation
- 03 GNN expressiveness: depth & scope
- 04 Conclusion & Opportunities**

Conclusion

- Graph learning is not only a buzzword in academia research, but also finds increasing impacts in industry
- When graph learning meets industry, vanilla models usually need modifications to meet the product needs
- When graph learning meets industry, novel frameworks must be developed to address the challenges in expressivity and scalability
- Speaker info: <https://research.facebook.com/people/xia-yinglong/>
- Open source: https://github.com/facebookresearch/shaDow_GNN

Opportunity



Yinglong Xia
yxia@meta.com

At Meta, we're connecting people to what they care about, powering new, meaningful experiences, and advancing the state-of-the-art through open research and accessible tooling. Our research on AI covers a variety of topics, ranging from ranking & recommendation, to user understanding, and to AI platform core technologies. Many well known & emerging AI techniques and tools, such as PyTorch and SparseNN model were born here.

Meta provides opportunities for FTE/STE, summer intern, visiting scholar, and university collaboration. We look forward to the possible collaboration.

Visiting Researcher, AI Mentorship Program,
Modern Recommendation Systems

