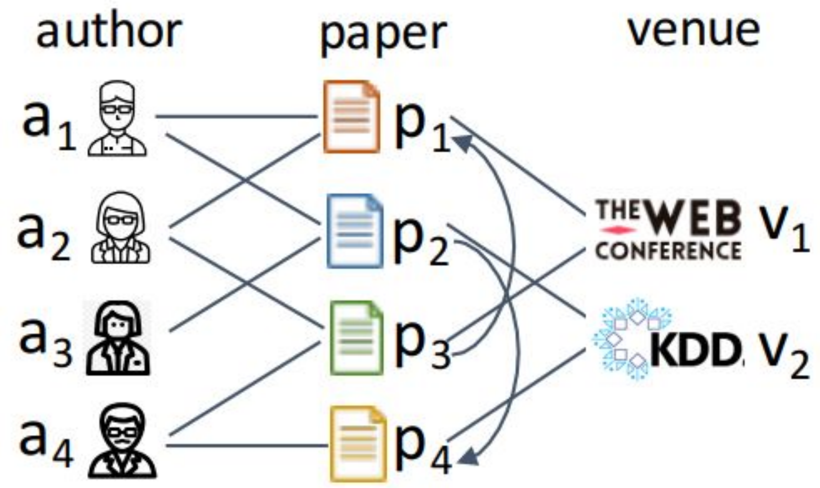
ActiveRGCN:
Finding Valuable Samples for Heterogeneous GNN Training

Xinyu Zhao, Haowei Jiang, Hang Zhang & Nuocheng Pan

UCLA Samueli School of Engineering

Heterogeneous Information Network (HIN)

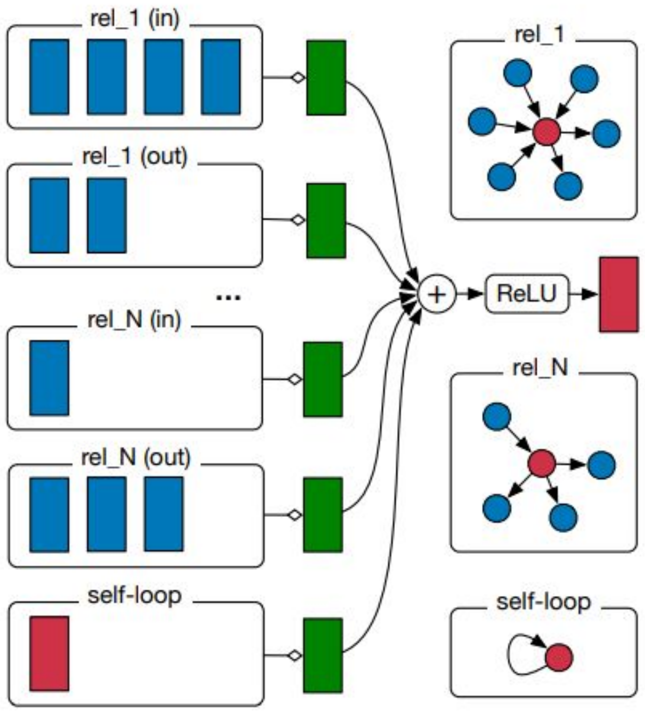
- Containing various types of nodes and edges (relationships)
- E.g.: Citation network
    - Node type: Papers, authors, conferences, …
    - Edge type: Author-Paper, Paper-Conference, Paper-paper, …
- Closer to more real world applications



Credit: picture from Heterogeneous Graph Neural Network

Relational Graph Convolutional Network

- An extension of GCN on HIN
- Compared with GCN
  - Directed Graph v.s. Undirected Graph
  - Aggregate neighbourhood information independently for each edge (considering relationship type and direction)
- Similar to GCN in homogeneous setting, RGCN is a very classic baseline for HIN network embedding.

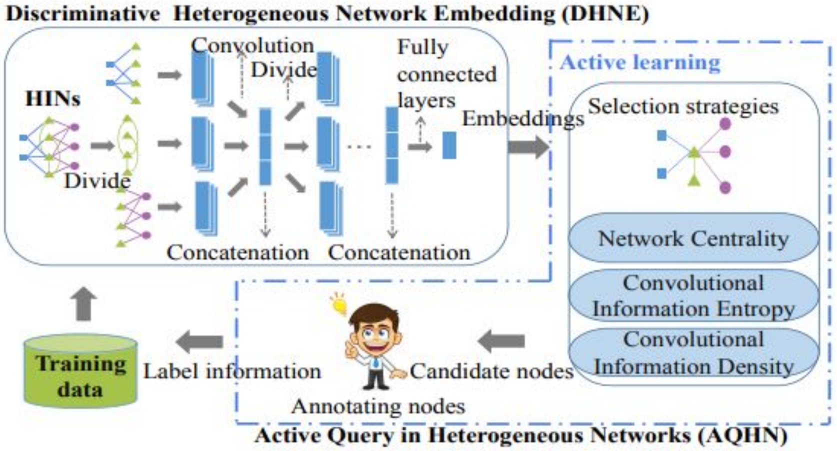


Credit: picture from [Modeling Relational Data with Graph Convolutional Networks]

UCLA Samueli School of Engineering

ActiveHNE

- An Active learning framework for HIN.
- Proposed a new heterogeneous network embedding methods.
- Proposed informative scored-based active learning framework together with a multi-armed bandit mechanism to dynamic update weight.



Credit: picture from [ActiveHNE: Active Heterogeneous Network Embedding]

UCLA Samueli School of Engineering

ActiveHNE Problem

- Experiment setting not fair enough
    - GCN as baseline while it is a homogeneous network

- Possible extension: Have active select working on other networks?

- Use Active Select on RGCN: **ActiveRGCN**

Active Select Strategy

- To select most representative nodes for learning without going through all
- Given Node ***v***, evaluate following 3 rewards:
  - Network Centrality (**NC**)
    - The representativeness of ***v***, measure degree
  - Convolutional Information Entropy (**CIE**)
    - Uncertainty of ***v***, by weighted sum of neighbor uncertainties
  - Convolutional Information Density (**CID**)
    - Representativeness of ***v*** in the embedding space based on neighbor nodes
- Weighted sum as a **reward function**
  - **Rewards** = **NC**_reward * **NC**_weight + **CIE**_reward * **CIE**_weight + **CID**_reward * **CID**_weight

UCLA Samueli School of Engineering

Reward Function Tuning

- An AL strategy inspired from Combinatorial **Multi-Arm Bandit (MAB)** problem
    - A RL method
    - Given a budget number of iterations
    - What player should do with an Arm, to maximize reward
    - Combinatorial allows to play multiple Arms during one iteration
    - Each arm corresponds to one of our rewards

- Dynamic update of the weights
- An estimation of the importance of a given node
- Select nodes by highest reward function with most change embedding: more new information

Selection Process: Pseudocodes

for ***ITER*** times:

Select ***BATCH*** nodes from **Training Set** have top **Rewards**

Add selected nodes to **Labeled Set**, remove them from **Training Set**

Create new **Model** (Model on previous ITER is destroyed)*

for ***EPOCH*** passes:

Train **Model** on **Labeled Set** once

Take **Embedding** from **Model**

Update **Weights** in **Rewards** from **Embedding** using **MAB**

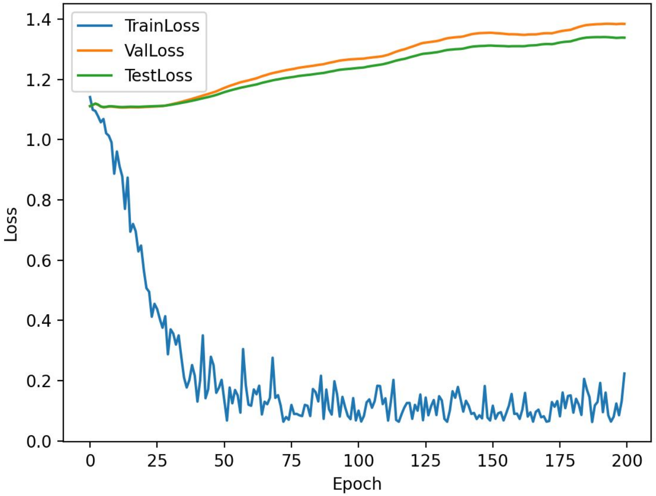
- Final **Labeled Set** have **Size=** ***ITER*** * ***BATCH***

Dataset & Setting

- ActiveHNE setting: epoch 200, iteration 40, batch size 20 for all dataset, save the last epoch result
- Our setting: epoch 50, iteration 40, batch size 80 for Cora, batch size 20 for DBLP and MovieLens, save the best model (lowest validation loss)

Dataset	Entities	Edges	Entity Type	Edge Type	class_num	**training set size**
Cora	56,670	244,088	3	3	10	**3911**
DBLP	37,791	170,803	4	3	4	**1014**
MovieLens	28,491	138,352	4	4	3	**918**

MovieLens

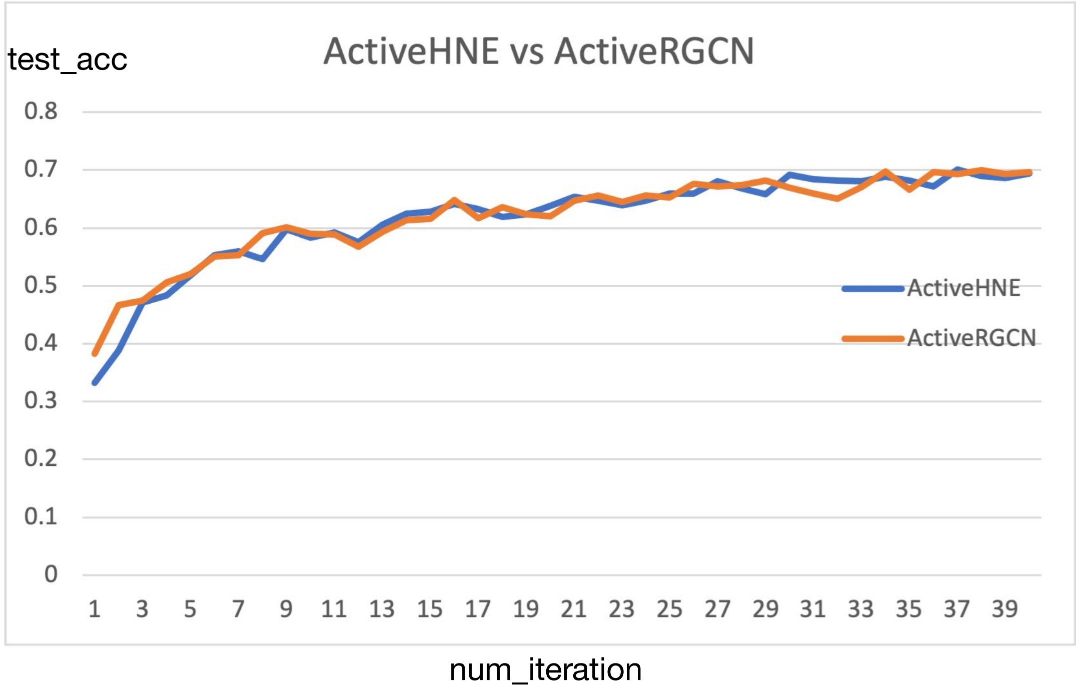


UCLA Samueli School of Engineering

Experiment Design

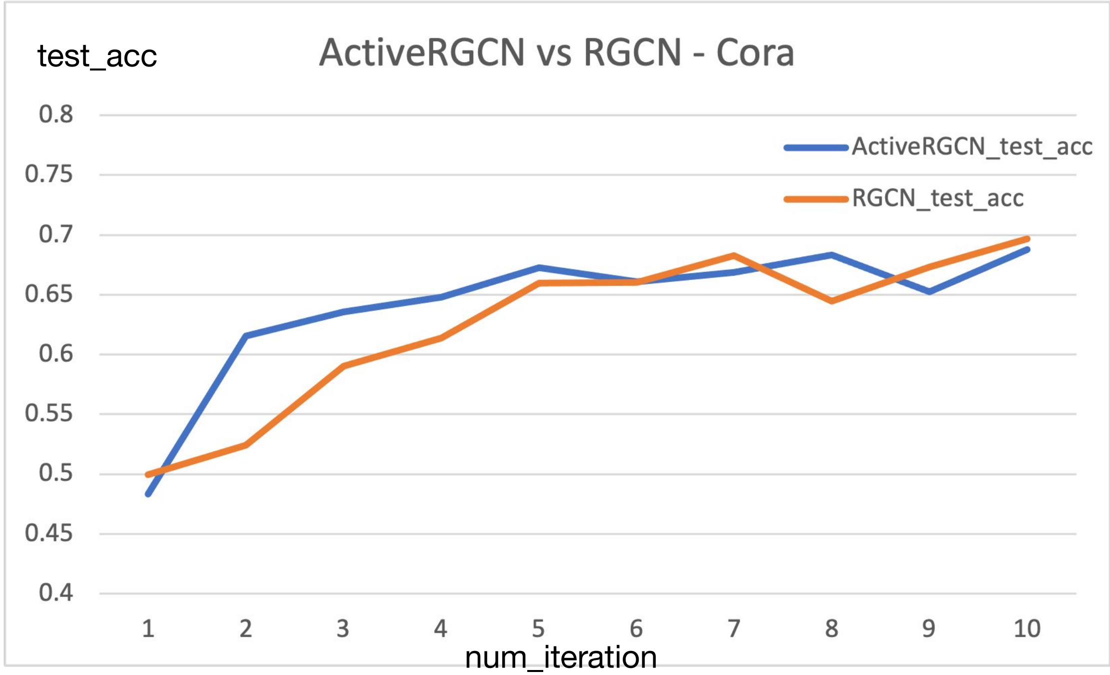
1. Apply active select to RGCN model, compare performance of ActiveHNE and ActiveRGCN , show the applicability of active select
2. Compare ActiveRGCN and RGCN
3. Ablation Study

ActiveHNE vs ActiveRGCN



- ActiveRGCN has simpler network structure, less training time, but similar performance to ActiveHNE
- Active select could be apply to other graph neural network, not only depend on DHNE or RGCN

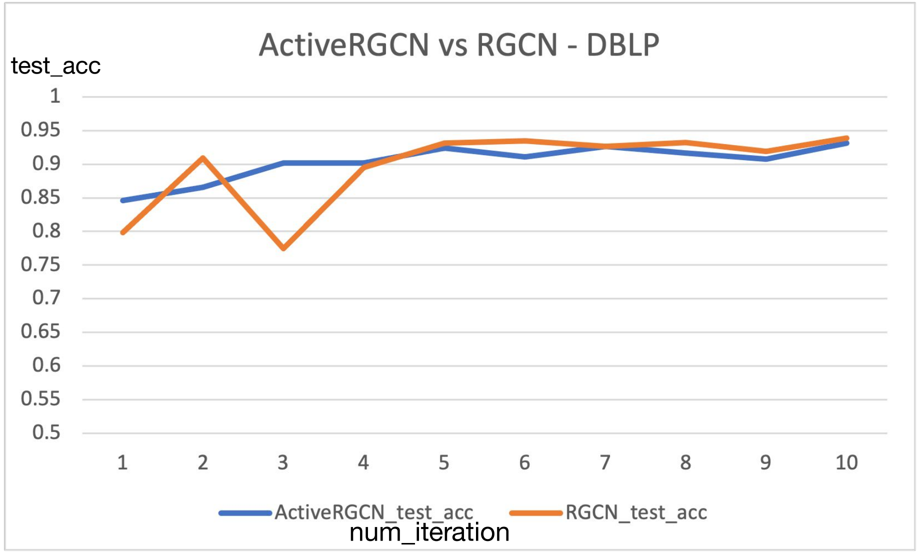
ActiveRGCN vs RGCN

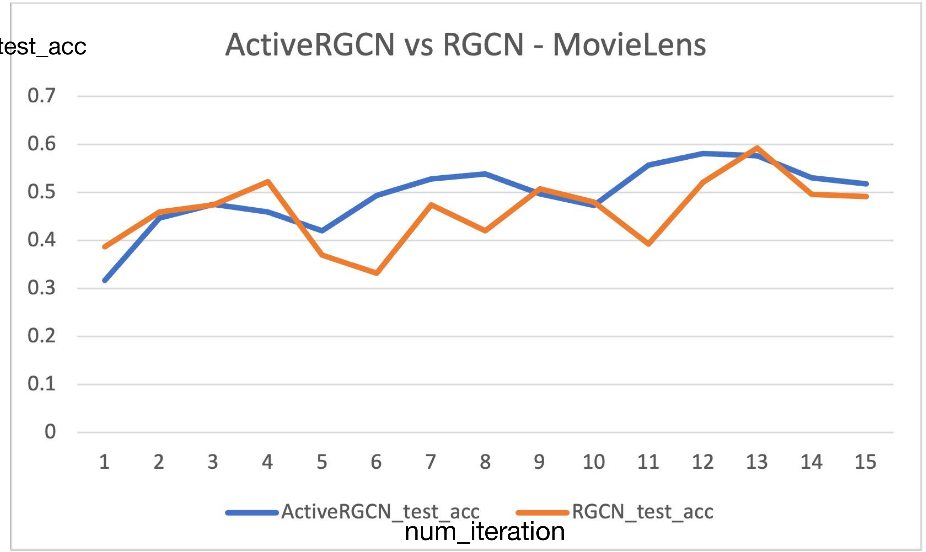


- Tests on **all three** Dataset
- Compare to RGCN, ActiveRGCN increase 5% accuracy for limited budget(less than 400 nodes), no significant difference in larger training set

UCLA Samueli School of Engineering

ActiveRGCN vs RGCN





UCLA Samueli School of Engineering

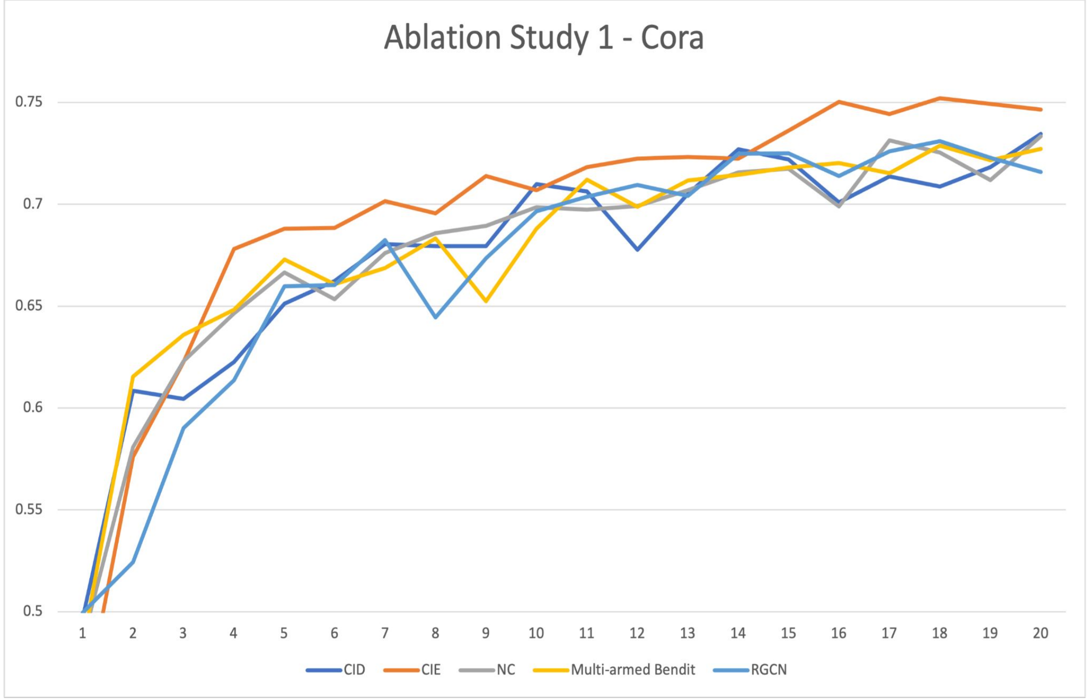
Ablation Study 1: effectiveness of each reward

- Active Learning Selection Strategy (Multi-armed Bendit mechanism):
    - **Rewards**[i] = **NC**_reward * **NC**_weight +

      CIE_reward * **CIE**_weight +

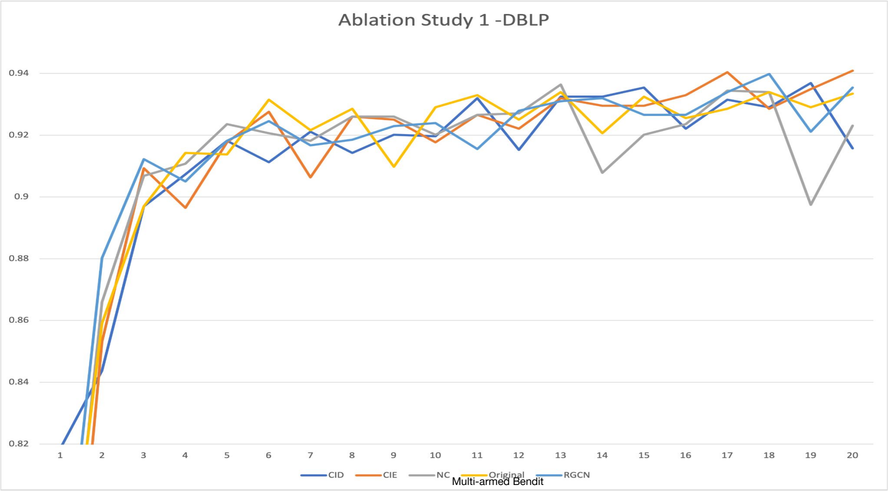
      CID_reward * **CID**_weight
- Among **three components**, which one is **most useful** for the Selection Strategy?
- We **tested each** component by **removing** the other two
    - **Rewards_NC**[i] = **NC**_reward * **NC**_weight
    - **Rewards_CIE**[i] = **CIE**_reward * **CIE**_weight
    - **Rewards_CID**[i] = **CID**_reward * **CID**_weight

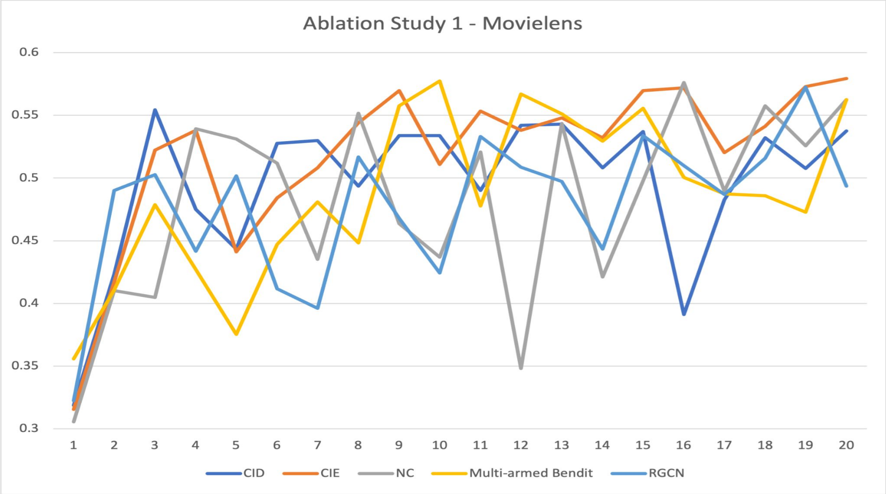
Ablation Study 1: effectiveness of each reward



- Tests on **Cora** Dataset
- All 4 tests used the same batch size and iteration number
- **CIE component curve** (orange line) has the **best accuracy** result, even better than the Multi-armed Bendit curve (yellow line)
- Compared to **RGCN baseline,** every reward is useful in finding valuable samples.

Ablation Study 1: effectiveness of each reward





- We did same experiments on **DBLP** and **MovieLens** Dataset
- **All rewards** are useful in finding informative nodes
- **CIE curve** (orange) has **least variance**, with relatively **best accuracy** result in three datasets
- The **Multi-armed Bendit setting curve** did **NOT** achieve the **best accuracy** result in general

UCLA Samueli School of Engineering

Ablation Study 1: effectiveness of each reward

- **Conclusion** of the first ablation study:
    - **CIE component contributes most** to the Active Learning Reward System.
    - **Each reward** is a **necessity** in the Reward System
    - **Multi-armed Bendit mechanism** does **NOT** assign a good **weight** to each reward
- This Leads to the **second ablation study**
    - We want to examine the **validity** of the Multi-armed Bendit mechanism.

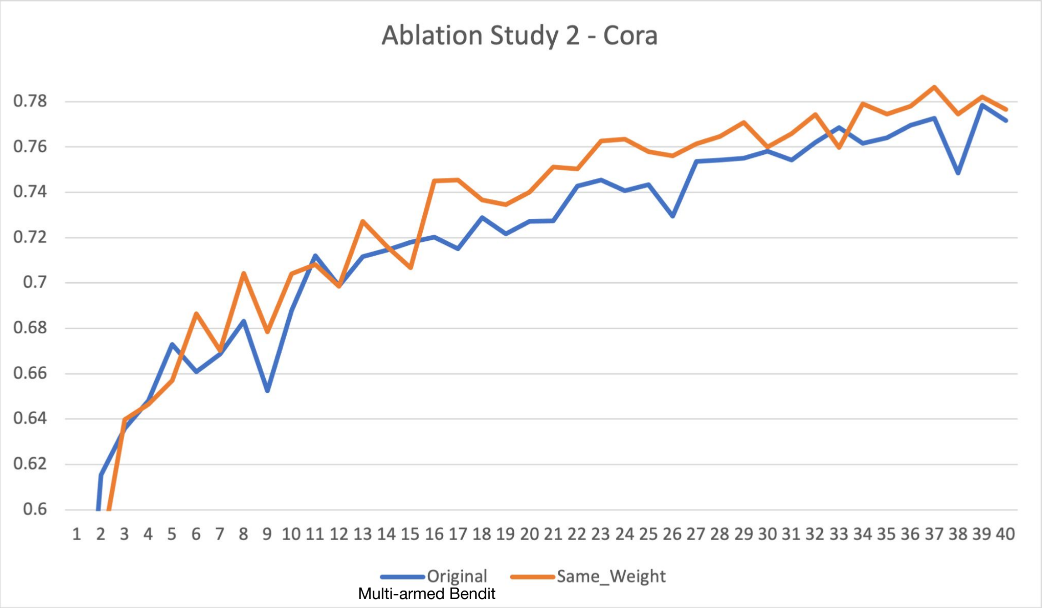
UCLA Samueli School of Engineering

Ablation Study 2: validity of multi-armed bandit

- **Multi-armed Bendit mechanism** updates weight of each reward dynamically:
  - **Rewards**[i] = **NC**_reward * **NC**_weight +
    CIE_reward * **CIE**_weight +
    CID_reward * **CID**_weight
- Is the **weight update** valid? Does it improve the performance of the Selection Strategy?
- We **tested** its weight update by **comparing** it with the constant **same-weight** setting:
  - **Rewards**[i] = **NC**_reward * 0.33 +
    CIE_reward * 0.33 +
    CID_reward * 0.33

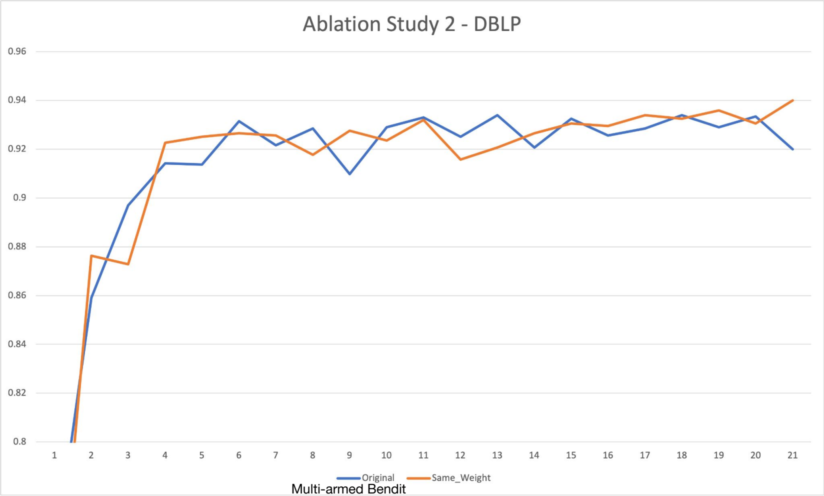
UCLA Samueli School of Engineering

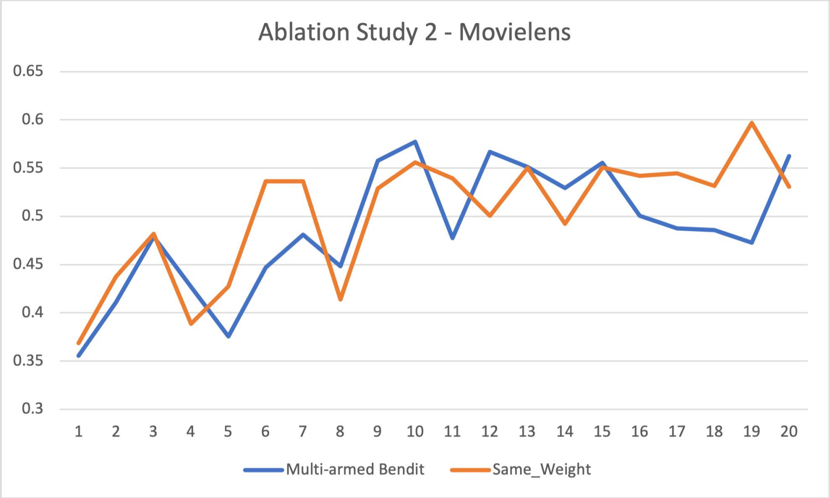
Ablation Study 2: validity of multi-armed bandit



- Tests on **Cora** Dataset
- Both two tests used the same batch size and iteration number
- Simple **Same-weight setting** (orange line) has **better accuracy** result than the multi-armed bendit setting curve (blue line)

Ablation Study 2: validity of multi-armed bandit





- We did same experiments on **DBLP** and **MovieLens** Dataset
- The **same-weight setting** curve has **less variance**, with relatively **better accuracy** result
- The **weight assignment** logic in the Multi-armed Bendit mechanism **needs to be improved**

Conclusion of our Project

- **Active Select Algorithm** is not model specific and can be applied to **other GNNs**
- **Compared to ActiveHNE**, **ActiveRGCN** uses a simple GNN to achieve the **same** performance
- **Under a specific budget, ActiveRGCN** has **a better** performance than the **RGCN**
- **Each reward** is useful in finding informative samples
- **CIE component** has the **most contribution** to the Active Select Reward system
- The **current Multi-armed Bendit mechanism** needs to be improved in weight updates

UCLA Samueli School of Engineering

Future Work

- **Wrap up** the current Active Learning Algorithm **into an API**
  - so that it can be **easily applied** to other GNNs, such as NARS and HAN
- Improve the **weight assignment logic** in Multi-armed Bendit mechanism
- Utilize the **one-step** Active Learning Algorithm in **GraphPart** paper
  - so that we can select all informative nodes **at once**
  - and **speed up** our Active Learning process

Q&A