KAIST

ICDM 2025
November 12–15, 2025
Washington DC, USA
IEEE International Conference on Data Mining

Identifying Group Anchors in Real-World Group Interactions Under Label Scarcity


Fanchen Bu


Geon Lee


Minyoung Choe

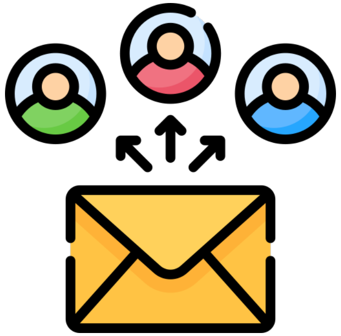

Kijung Shin

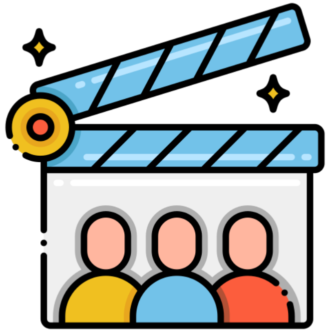
Group Interactions Are Everywhere

MAN IS BY NATURE A SOCIAL ANIMAL.

- Aristotle (384 – 322 BC; Ancient Greek Philosopher)

- **Group interactions** are a fundamental part of our world
- **Co-authorship:** Scholars collaborate on a research paper
- **Online Q&A:** A user posts a question and others join in to answer
- **Email/Social-media messages:** A user sends a message to others
- **Movie cast:** Actors perform together in a film



Observation: Anchors in Group Interactions

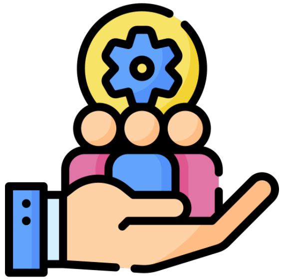
EVERY FRIEND GROUP HAS THAT ONE PERSON WHO KEEPS EVERYONE TOGETHER.

- Anonymous Redditor

- In each group interaction, there is often an "anchor", a particularly important person that brings together the group members
- **Co-authorship:** The *first/last author* of a paper
- **Online Q&A:** The *questioner* who posts a question
- **Email/Social-media messages:** The *sender* who sends a message
- **Movie cast:** The *leading actor* in a film
- In this work, we study **how we can identify anchors in real-world group interactions**

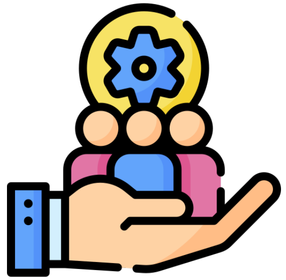
Group Anchor Identification: Applications

- **Group interaction prediction:** Anchors often initiate group formation
  → Identifying them helps predict future groups
  - E.g., future academic/business collaborations
- **Engagement management:** Anchors often play important roles
  → Understanding them helps maintain group health and activity
  - E.g., Social-media community management
- **Targeted marketing:** Anchors are influencers within their groups
  → Reaching them can be more effective for marketing
  - E.g., product seeding and influencer marketing

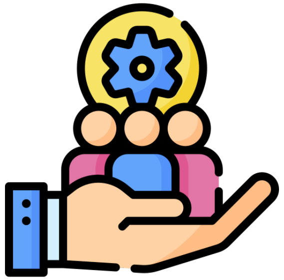

Group Anchor Identification: Applications

- **Group interaction prediction:** Anchors often initiate group formation
  → Identifying them helps predict future groups
  - E.g., future academic/business collaborations
- **Engagement management:** Anchors often play important roles
  → Understanding them helps maintain group health and activity
  - E.g., Social-media community management
- **Targeted marketing:** Anchors are influencers within their groups
  → Reaching them can be more effective for marketing
  - E.g., product seeding and influencer marketing

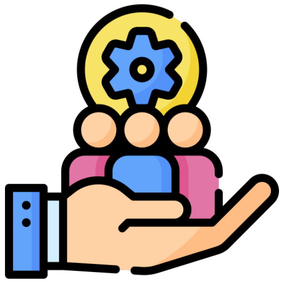

Group Anchor Identification: Applications

- **Group interaction prediction:** Anchors often initiate group formation
  → Identifying them helps predict future groups
  - E.g., future academic/business collaborations
- **Engagement management:** Anchors often play important roles
  → Understanding them helps maintain group health and activity
  - E.g., Social-media community management
- **Targeted marketing:** Anchors are influencers within their groups
  → Reaching them can be more effective for marketing
  - E.g., product seeding and influencer marketing



Group Anchor Identification: Applications

- **Group interaction prediction:** Anchors often initiate group formation
  → Identifying them helps predict future groups
  - E.g., future academic/business collaborations
- **Engagement management:** Anchors often play important roles
  → Understanding them helps maintain group health and activity
  - E.g., Social-media community management
- **Targeted marketing:** Anchors are influencers within their groups
  → Reaching them can be more effective for marketing
  - E.g., product seeding and influencer marketing



Group Anchor Identification: Problem Statement

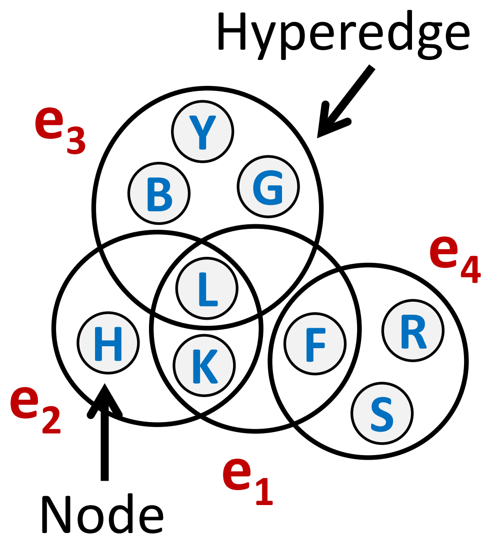
- We formulate it as an optimization problem on **hypergraphs**
- **Hypergraph:** $H = (V, E)$ with node set V and hyperedge set E
- A node = a person; A hyperedge = a group interaction among people
- Below is an example of co-authorship hypergraph

Authors (Nodes)

Jure Leskovec (**L**)	Austin Benson (**B**)
Jon Kleinberg (**K**)	David Gleich (**G**)
Hao Yin (**Y**)	Timos Sellis (**S**)
Christos Faloutsos (**F**)	Nick Roussopoulos (**R**)
Daniel Huttenlocher (**H**)	

Publications (Hyperedges)

e_1: (L, K, F) KDD'05
e_2: (L, H, K) WWW'10
e_3: (Y, B, G, L) KDD'17
e_4: (S, R, F) VLDB'87



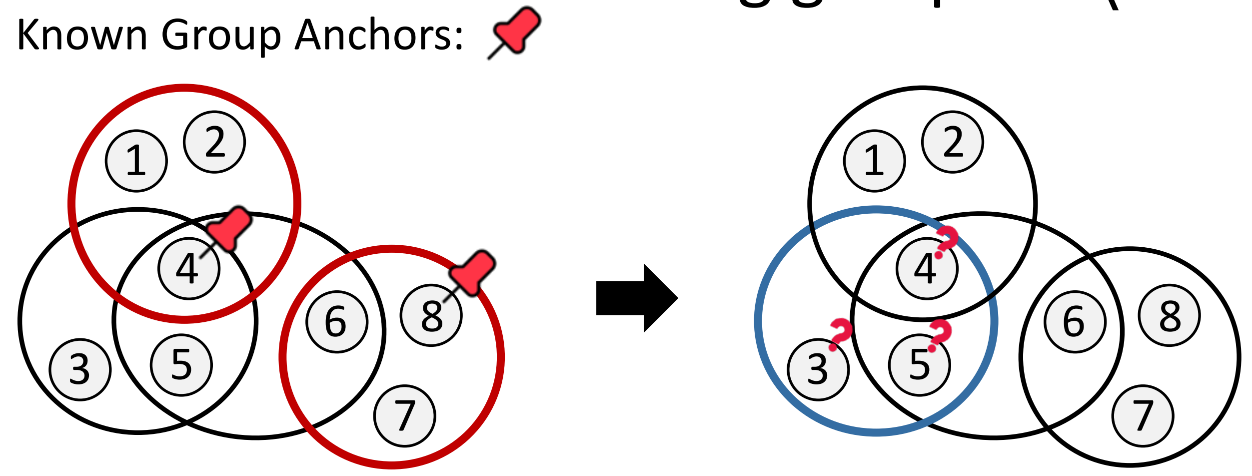
Group Anchor Identification: Problem Statement

- We introduce the concepts of **domains** and **anchor roles**
- We consider real-world hypergraphs, each with a known *domain* $\mathcal{D}$
- For each domain, we identify its *anchor role* $\mathcal{R}(\mathcal{D})$, the role of the anchor in each group in that domain
  - For the co-authorship domain, for each paper, either the first or last author is arguably the anchor, and we consider both alternative cases

Domain $\mathcal{D}$	Nodes	Anchor Role $\mathcal{R}(\mathcal{D})$
$\mathcal{D}_{\text{co}}$: Co-authorship	Authors of a paper	First/last author
$\mathcal{D}_{\text{qa}}$: Online Q&A	Users involved in a question	Questioner
$\mathcal{D}_{\text{em}}$: Email	People involved in an email	Sender
$\mathcal{D}_{\text{so}}$: Social network	Users involved in a communication	Initiator
$\mathcal{D}_{\text{mo}}$: Movie cast	Actors performing in a movie	Leading actor

Group Anchor Identification: Problem Statement

- **Given:** (1) A real-world hypergraph $H = (V, E)$ and (2) known anchors in some groups $E' \subseteq E$
  - In each $e' \in E'$, we know the node $v' \in e'$ that has the anchor role $\mathcal{R}(\mathcal{D})$
  - **Label scarcity:** We consider the realistic scenarios where the proportion of groups with known anchors is limited
- **To predict:** The anchors in the remaining groups $E \setminus E'$

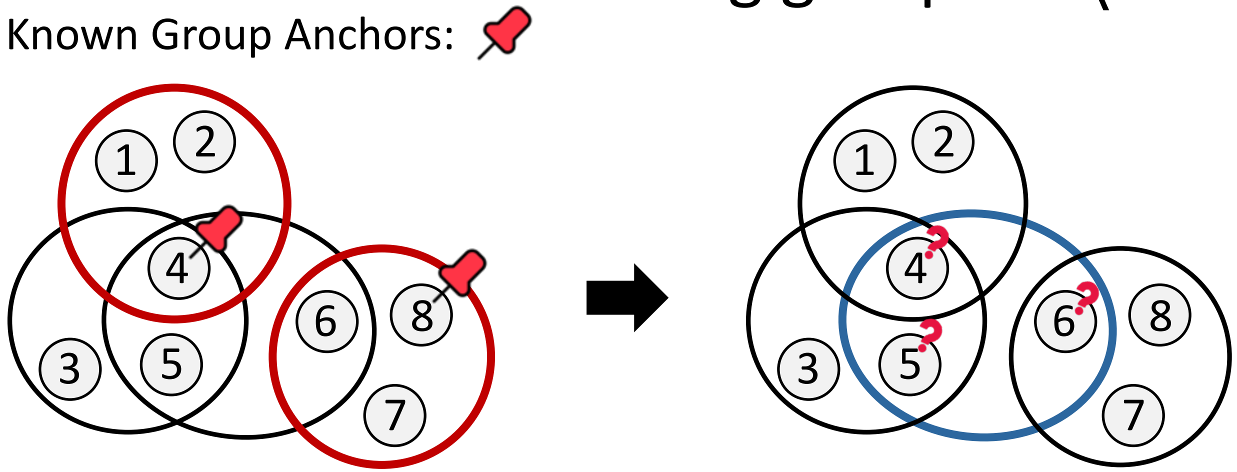


A Hypergraph with Known Anchors in Some Groups

Predict the Anchor in Each Remaining Group

Group Anchor Identification: Problem Statement

- **Given:** (1) A real-world hypergraph $H = (V, E)$ and (2) known anchors in some groups $E' \subseteq E$
  - In each $e' \in E'$, we know the node $v' \in e'$ that has the anchor role $\mathcal{R}(\mathcal{D})$
  - **Label scarcity:** We consider the realistic scenarios where the proportion of groups with known anchors is limited
- **To predict:** The anchors in the remaining groups $E \setminus E'$

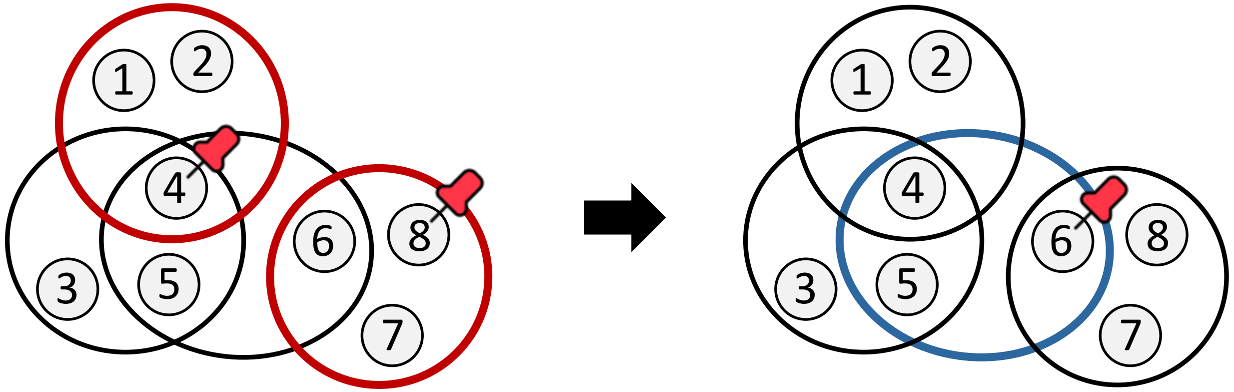


A Hypergraph with Known Anchors in Some Groups

Predict the Anchor in Each Remaining Group

Group Anchor Identification: Group-Dependence

- The group anchors are *group-dependent*
- E.g., 4 is the anchor in the group $\{1,2,4\}$ does NOT necessarily mean 4 is also the anchor in other groups such as $\{3,4,5\}$ and $\{4,5,6\}$
- Similarly, 6 is a non-anchor in the group $\{6,7,8\}$ does NOT necessarily mean 6 cannot be the anchor of other groups such as $\{4,5,6\}$
- In the example, it is possible that 6 is the anchor in the group $\{4,5,6\}$

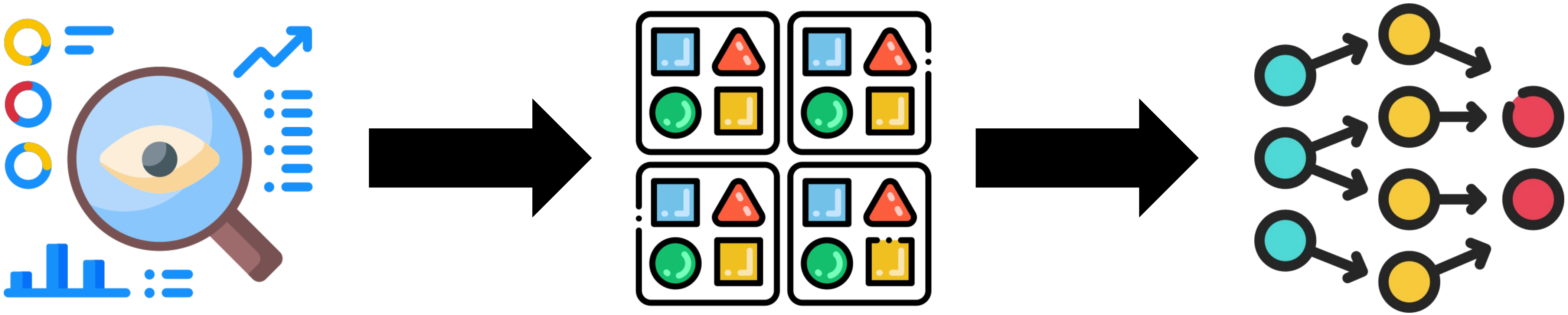


A Hypergraph with Known Anchors in Some Groups

Predict the Anchor in Each Remaining Group

High-Level Idea: Observation-Driven Approach

- **Idea:** Instead of using a sophisticated "black-box" model, we first observe patterns in the real-world group interaction data, and then design a lightweight method based on those insights
- **Why is this a good approach for this problem?**
  - **Well handles label scarcity:** With very little training data, complex models can fail, while a lightweight, observation-driven model is more robust
  - **Intuitive and interpretable:** The final method is easy to understand because it's directly motivated by real-world patterns



Observations: Settings

- Since we consider **label scarcity** in our problem, we also impose this constraint for our observations
  - We establish our observations and the patterns with the same proportion (7.5%) of known anchors as in our main experiments
- We assume **no node or edge attributes** (i.e., external features) are given, which is true for the real-world datasets used in this work
  - That is, we only have information from (1) the *hypergraph topology* and (2) the label information of the *known group anchors*

Observations: Datasets

- We use 13 datasets from 5 different domains

Domain	Dataset	Abbrev.	|V|	|E|	|E*|	Min. |e|	Max. |e|	Avg. |e|
Co-authorship ($\mathcal{D}_{\text{co}}$)	AMinerAuthor [21], [22]	coAA	1,712,433	2,037,605	1,454,250	1	115	2.55
	DBLP [22], [23]	coDB	108,476	91,260	81,601	2	36	3.52
	ScopusMultilayer [24]–[27]	coSM	1,673	937	842	1	27	3.09
Online Q&A ($\mathcal{D}_{\text{qa}}$)	StackOverflowBiology [22]	qaBI	15,418	26,290	23,242	1	12	2.08
	StackOverflowPhysics [22]	qaPH	80,434	194,575	169,274	1	40	2.38
	MathOverflow [24]	qaMA	410	154	154	2	57	4.27
	StackOverflow [24]	qaST	22,131	4,716	4,713	1	59	5.79
Email ($\mathcal{D}_{\text{em}}$)	EmailEnron [22]	emEN	21,251	101,124	34,916	2	883	11.53
	EmailEu [22], [28]	emEU	986	209,508	24,520	2	40	2.56
	Enron [24]	emER	110	9,603	1,169	2	29	2.47
Social network ($\mathcal{D}_{\text{so}}$)	Message [24]	soME	26,059	34,577	22,700	2	14	2.58
	Retweet [24]	soRE	30,073	88,148	49,828	2	2	2.00
Movie cast ($\mathcal{D}_{\text{mo}}$)	MovieLens [24], [29]	moML	73,155	43,058	42,497	1	5	4.70

*Data source: https://github.com/young917/EdgeDependentNodeLabel [22] and https://andrewmellor.co.uk/data/ [24].

Observation 1: Informative Topological Features

- **Recall:** We only have information from (1) the *hypergraph topology* and (2) the label information of the *known group anchors*
- **Observation 1 focuses on part (1):** What can the topology tell us?
- Topology → Topological features → But are they helpful?

Observation 1: Informative Topological Features

- **Recall:** We only have information from (1) the *hypergraph topology* and (2) the label information of the *known group anchors*
- **Observation 1 focuses on part (1):** What can the topology tell us?
- Topology → Topological features → But are they helpful?

Observation 1: Informative Topological Features

We can identify group anchors *fairly accurately* using *only topological features*.

- **What is the intuition behind this observation?**
- Let's consider one of the simplest topological features: *node degree*
- **Co-authorship:** High degree → Senior scholar → Likely last author
- **Online Q&A:** Low degree → New user → Likely questioner
- **Movie cast:** High degree → Famous actor → Likely leading actor

Observation 1: Informative Topological Features

We can identify group anchors *fairly accurately* using *only topological features*.

- **What evidence do we have?**
- We report the accuracy of predicting the highest- or lowest-degree node in each group as the anchor, and compare it with SOTA baselines
- This simple method shows *considerable* performance!

Dataset	Degree	WHATsNet	CoNHD-U	CoNHD-I	Random
coAA (first)	44.4	**45.2**	42.7	44.5	37.1
coAA (last)	**46.3**	45.8	42.2	44.7	37.1
coDB (first)	**42.5**	42.5	41.2	41.4	32.8
coDB (last)	**45.5**	45.4	42.7	43.5	32.8
coSM (first)	32.7	**34.3**	31.4	29.8	33.7
coSM (last)	37.7	**39.8**	37.9	39.4	33.7
qaBI	76.2	**85.6**	78.7	79.3	44.3
qaPH	77.2	**88.1**	76.0	77.3	41.1
qaMA	**38.7**	35.8	29.0	29.8	32.4
qaST	30.9	**31.2**	25.4	26.6	24.2
emEN	22.0	**50.8**	44.0	45.1	18.8
emEU	49.0	**51.0**	52.8	52.4	45.8
emER	66.2	**66.6**	65.3	64.6	44.9
soME	65.0	**75.5**	74.3	74.6	42.9
soRE	84.3	97.4	96.8	**97.5**	50.0
moML	41.8	41.4	42.4	**42.7**	21.3
Avg. Acc.	50.0	**54.8**	51.4	52.1	35.8
Avg. Rank	2.75	**1.63**	3.38	2.56	4.69

Observation 2: Stable Cross-Group Anchorship

- **Recall:** We only have information from (1) the *hypergraph topology* and (2) the label information of the *known group anchors*
- **Observation 2 focuses on part (2):** What can the known group anchors tell us?
- Anchors are group-dependent → But can we still observe any correlations between the anchorship in different groups?

Observation 2: Stable Cross-Group Anchorship

- **Recall:** We only have information from (1) the *hypergraph topology* and (2) the label information of the *known group anchors*
- **Observation 2 focuses on part (2):** What can the known group anchors tell us?
- Anchors are group-dependent → But can we still observe any correlations between the anchorship in different groups?

Observation 2: Stable Cross-Group Anchorship

If a node is (not) the anchor in some groups,
it is likely (not) the anchor in other groups too.

- **What is the intuition behind this observation?**
- **Co-authorship:** Last author of several papers → Usually professor → Likely the last author of other papers
- **Email:** Sender of several emails → Maybe in charge of announcement → Likely the sender of other emails
- **Movie cast:** Leading actor in several movies → Maybe famous movie star → Likely the leading actor in other movies

Observation 2: Stable Cross-Group Anchorship

If a node is (not) the anchor in some groups,
it is likely (not) the anchor in other groups too.

- **What evidence do we have?**
- **Anchor purity of node v:** When we randomly pick two groups containing v, the probability that v is the anchor in *both or neither* of the two groups
- The average anchor purity in real-world group interactions v.s. randomized ones → It is much higher in real-world group interactions!

Dataset		Real-world	Random	p-value
coAA	(first)	0.7420±0.3706	0.5762±0.4012	<0.0001
	(last)	0.7375±0.3662	0.5758±0.4012	<0.0001
coDB	(first)	0.7708±0.3786	0.5873±0.4242	<0.0001
	(last)	0.7490±0.3801	0.5977±0.4209	<0.0001
coSM	(first)	0.7821±0.3777	0.5103±0.4728	0.0146
	(last)	0.8872±0.2335	0.5577±0.4598	0.0012
qaBI		0.8196±0.3248	0.5323±0.3692	<0.0001
qaPH		0.8146±0.3239	0.5375±0.3700	<0.0001
qaMA		0.8750±0.3307	0.6250±0.4841	0.1391
qaST		0.9051±0.2696	0.7551±0.3987	0.0141
emEN		0.9430±0.1700	0.8551±0.2408	<0.0001
emEU		0.6501±0.2217	0.5842±0.1941	<0.0001
emER		0.7890±0.2499	0.6014±0.2331	<0.0001
soME		0.6872±0.4048	0.6701±0.4138	0.0834
soRE		0.7268±0.3713	0.5498±0.3790	<0.0001
moML		0.9962±0.0494	0.5077±0.3285	<0.0001
Avg. Purity		0.8034	0.6015	-

Observation 2: Stable Cross-Group Anchorship

If a node is (not) the anchor in some groups,
it is likely (not) the anchor in other groups too.

- Observation 2 tells us the *cross-group stability* of anchorship
- **What mechanism is possibly behind this observation?**
- We hypothesize that each node v has a *global anchor strength* shared *across all groups* involving v
- The global anchor strength of v indicates the overall likelihood of v being the anchor across different groups

Observation 2: Stable Cross-Group Anchorship

If a node is (not) the anchor in some groups,
it is likely (not) the anchor in other groups too.

- **Hypothesis:** Each node v has a *global anchor strength*
- **What evidence do we have?**
- **Anchor proportion of node v:** The proportion of the groups where v is the anchor, among all the groups involving v
- The proportion of groups where the node with the highest anchor proportion is the anchor is very high! → Such global strengths exist!

Dataset	coAA (first)	coAA (last)	coDB (first)	coDB (last)	coSM (first)	coSM (last)	qaBI	qaPH	qaMA	qaST	emEN	emEU	emER	soME	soRE	moML	Avg.
Acc. (%)	93.5	92.9	97.1	96.1	98.3	100.0	98.9	98.6	100.0	100.0	80.6*	59.4*	81.3*	92.8	88.8	100.0	92.4

*Email datasets, especially emEU, contain many repeated hyperedges consisting of the same nodes but with different anchors

Observation 2: Stable Cross-Group Anchorship

If a node is (not) the anchor in some groups,
it is likely (not) the anchor in other groups too.

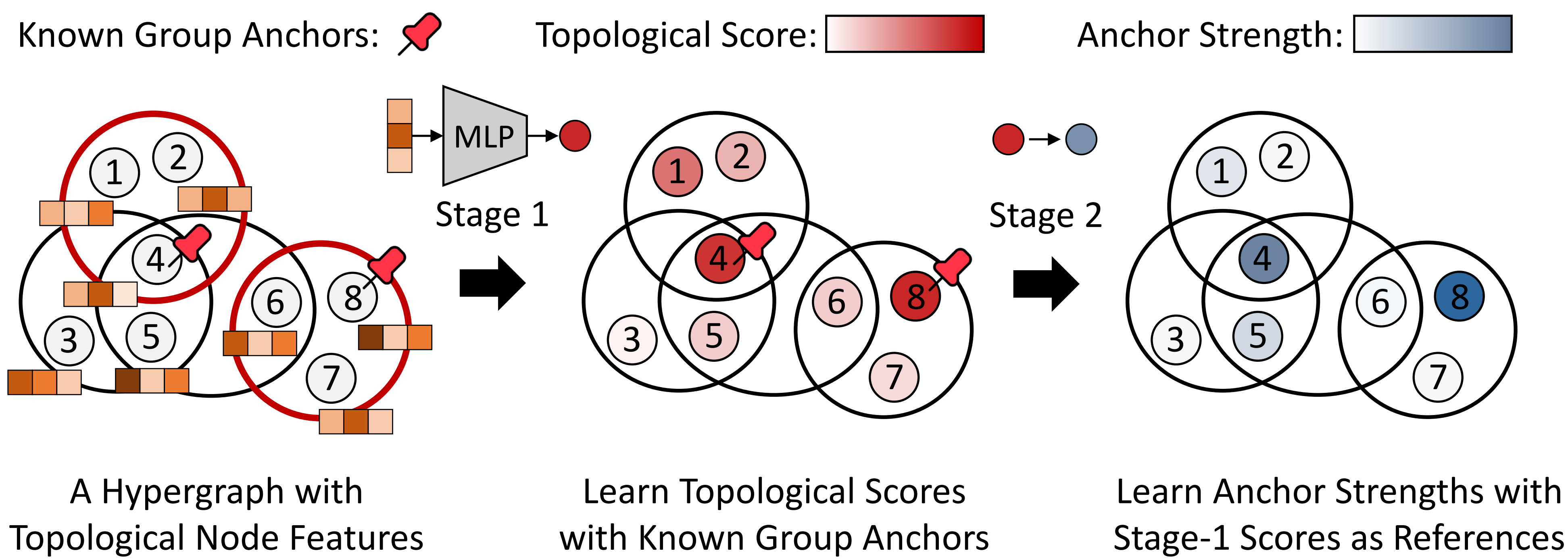
- The proportion of groups where the node with the highest anchor proportion is the anchor is very high! → Such global strengths exist!
- **This is not a “method”!** This is only used to *validate our hypothesis* that there exist global strengths that can well-explain the anchorship

Dataset	coAA (first)	coAA (last)	coDB (first)	coDB (last)	coSM (first)	coSM (last)	qaBI	qaPH	qaMA	qaST	emEN	emEU	emER	soME	soRE	moML	Avg.
Acc. (%)	93.5	92.9	97.1	96.1	98.3	100.0	98.9	98.6	100.0	100.0	80.6*	59.4*	81.3*	92.8	88.8	100.0	92.4

*Email datasets, especially emEU, contain many repeated hyperedges consisting of the same nodes but with different anchors

Proposed Method AnchorRadar: Overview

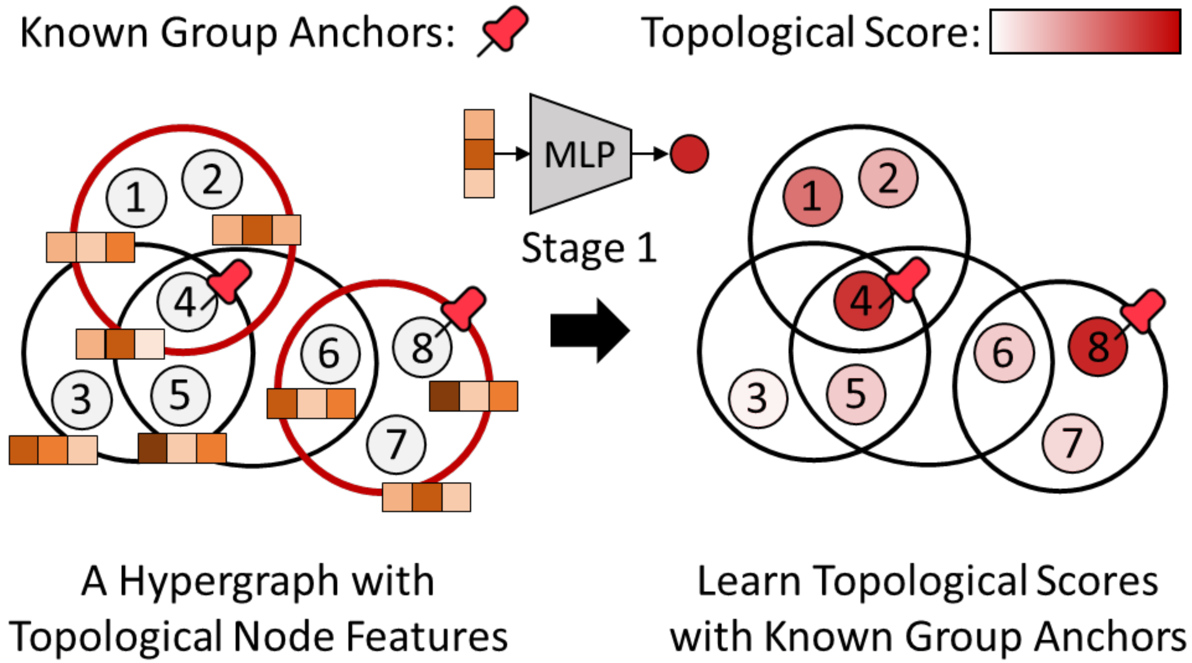
- The proposed method AnchorRadar has two stages
  - Stage 1 is based on observation 1, and Stage 2 is based on observation 2
- **Stage 1:** Train an MLP to learn *topological scores* to fit known anchors
- **Stage 2:** Train *anchor strengths* with Stage-1 scores as references



Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

- Train a model to exploit the correlations between topological features and anchorship
- Use a **lightweight** architecture
  - Specifically, MLP
- Use **topological features** *as the* **only inputs** to fit known anchors



Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

Algorithm 1: AnchorRadar: Stage 1

Input: (1) V and E: topology; (2) E' and $A(e), \forall e \in E'$: known group anchors; (3) X: topological feature matrix; (4) $n_{ep}^{(1)}$: number of optimization epochs

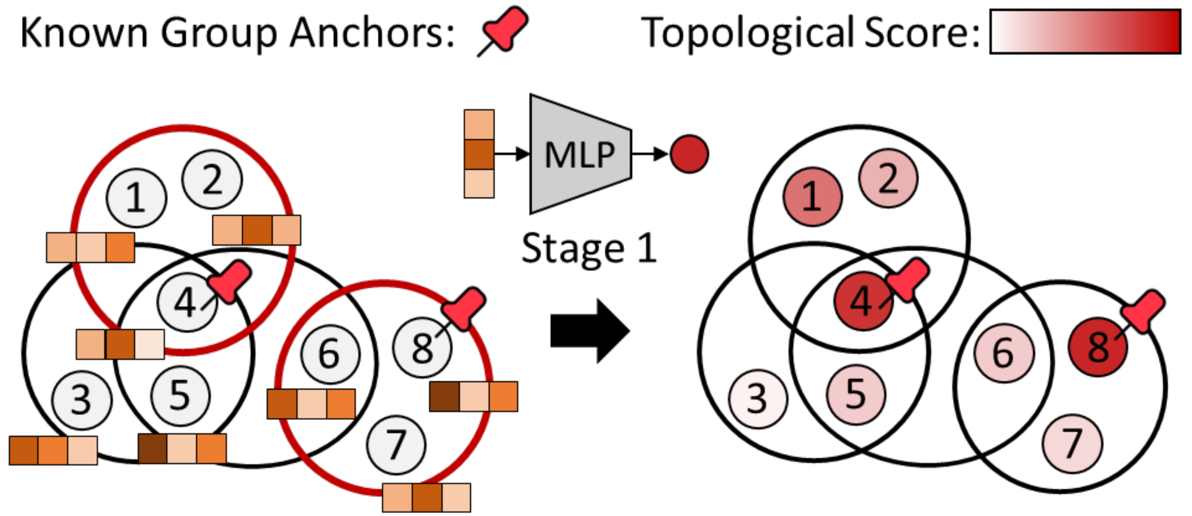
Output: $s_{v;e}^{(1)}, \forall v \in e \in E$: learned topology-based scores

for $i_{ep} = 1, 2, \ldots, n_{ep}^{(1)}$ **do**

- $s_{v;e}^{(1)} = \mathrm{MLP}(X;\theta)$ ▷ MLP forward pass
- $\mathcal{L}^{(1)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}$ ▷ Eq. (1)
- Update θ w.r.t. $\frac{\partial \mathcal{L}^{(1)}}{\partial \theta}$ ▷ Gradient descent

return $s_{v;e}^{(1)} = \mathrm{MLP}(X;\theta)$ ▷ Trained scores

---



A Hypergraph with Topological Node Features

Learn Topological Scores with Known Group Anchors

- We followed existing works, using (1) node degree, (2) eigenvector centrality, (3) PageRank centrality, and (4) coreness

Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

Algorithm 1: AnchorRadar: Stage 1

Input: (1) V and E: topology; (2) E' and $A(e), \forall e \in E'$: known group anchors; (3) X: topological feature matrix; (4) $n_{ep}^{(1)}$: number of optimization epochs

Output: $s_{v;e}^{(1)}, \forall v \in e \in E$: learned topology-based scores

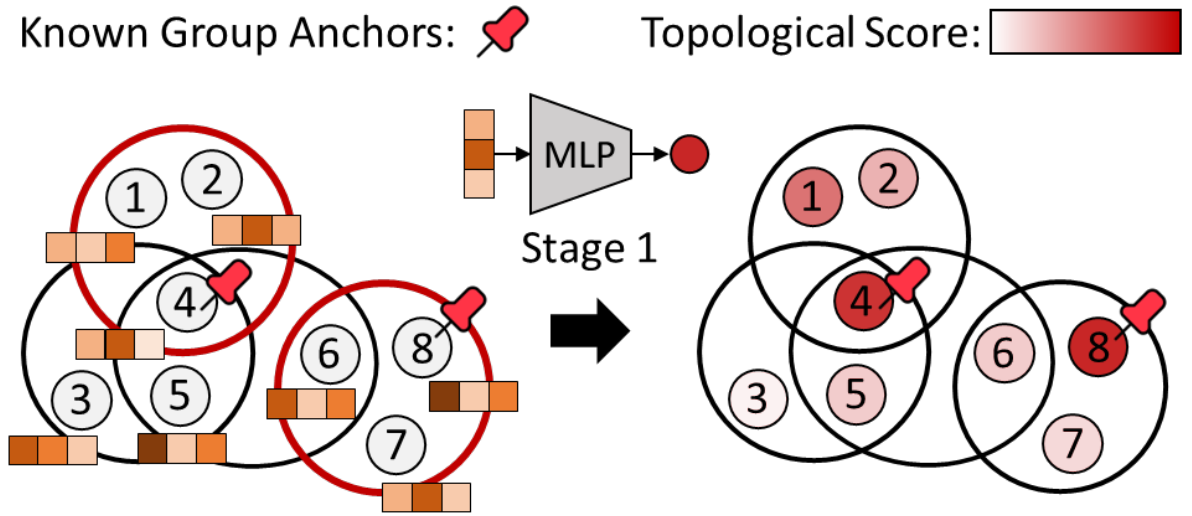
for $i_{ep} = 1, 2, \ldots, n_{ep}^{(1)}$ **do**

  $s_{v;e}^{(1)} = \mathrm{MLP}(X; \theta)$ ▷ MLP forward pass

  $\mathcal{L}^{(1)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}$ ▷ Eq. (1)

  Update θ w.r.t. $\frac{\partial \mathcal{L}^{(1)}}{\partial \theta}$ ▷ Gradient descent

return $s_{v;e}^{(1)} = \mathrm{MLP}(X; \theta)$ ▷ Trained scores



A Hypergraph with Topological Node Features

Learn Topological Scores with Known Group Anchors

- We build X using both hypergraph-level and group-level aggregations and normalizations → A feature vector for each node-group pair (v, e)

Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

Algorithm 1: AnchorRadar: Stage 1

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** X: topological feature matrix; **(4)** $n_{ep}^{(1)}$: number of optimization epochs

Output: $s_{v;e}^{(1)}, \forall v \in e \in E$: learned topology-based scores

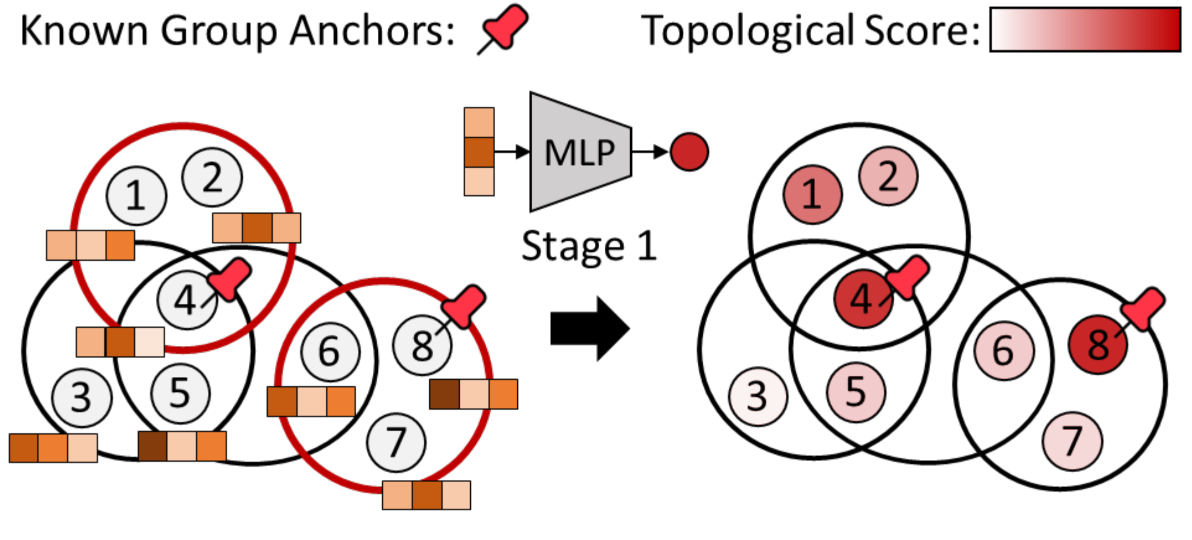
for $i_{ep} = 1, 2, \ldots, n_{ep}^{(1)}$ **do**

  $\boxed{s_{v;e}^{(1)} = \text{MLP}(X; \theta)}$ ▷ MLP forward pass

  $\mathcal{L}^{(1)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}$ ▷ Eq. (1)

  Update θ w.r.t. $\frac{\partial \mathcal{L}^{(1)}}{\partial \theta}$ ▷ Gradient descent

return $s_{v;e}^{(1)} = \text{MLP}(X; \theta)$ ▷ Trained scores



A Hypergraph with Topological Node Features

Learn Topological Scores with Known Group Anchors

- Each node-group pair (v, e) has its *topological score* $s_{v;e}^{(1)}$
  → Higher = the node v is more likely the anchor in the group e

Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

Algorithm 1: AnchorRadar: Stage 1

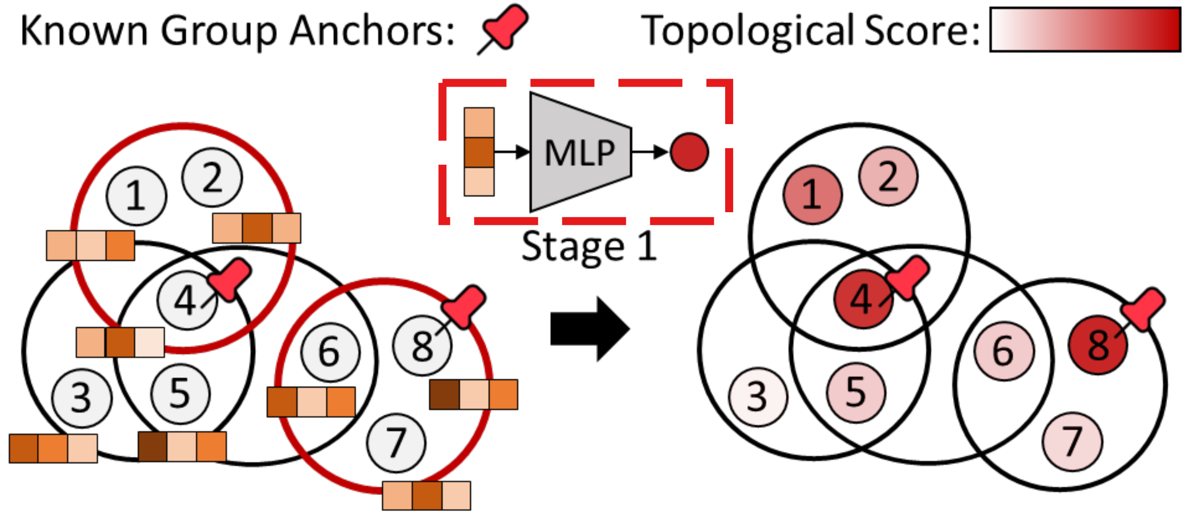
Input: (1) V and E: topology; (2) E' and $A(e), \forall e \in E'$: known group anchors; (3) X: topological feature matrix; (4) $n_{ep}^{(1)}$: number of optimization epochs

Output: $s_{v;e}^{(1)}, \forall v \in e \in E$: learned topology-based scores

for $i_{ep} = 1, 2, \ldots, n_{ep}^{(1)}$ **do**

- $\boxed{s_{v;e}^{(1)} = \text{MLP}(X;\theta)}$ ▷ MLP forward pass
- $\mathcal{L}^{(1)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}$ ▷ Eq. (1)
- Update θ w.r.t. $\frac{\partial \mathcal{L}^{(1)}}{\partial \theta}$ ▷ Gradient descent

return $s_{v;e}^{(1)} = \text{MLP}(X;\theta)$ ▷ Trained scores



A Hypergraph with Topological Node Features

Learn Topological Scores with Known Group Anchors

- Each topological score $s_{v;e}^{(1)}$ is computed from topological features X transformed by an MLP

Proposed Method AnchorRadar: Stage 1

Observation 1: In real-world group interactions, topological features are informative about group anchors.

Algorithm 1: AnchorRadar: Stage 1

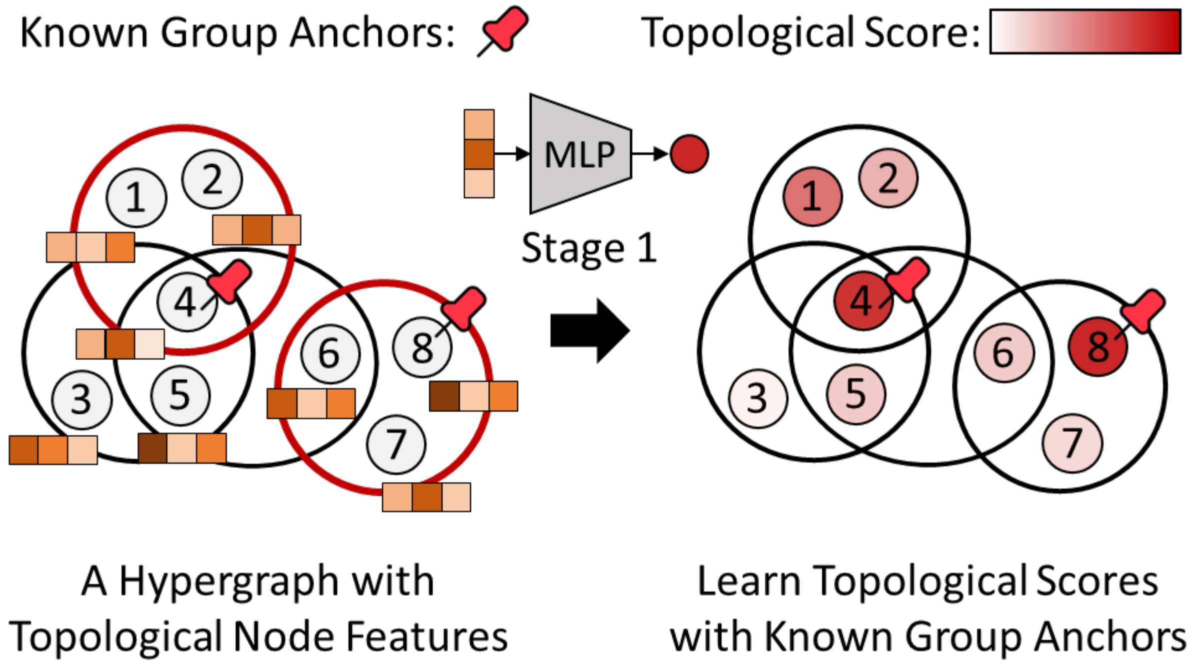
Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** X: topological feature matrix; **(4)** $n_{ep}^{(1)}$: number of optimization epochs

Output: $s_{v;e}^{(1)}, \forall v \in e \in E$: learned topology-based scores

for $i_{ep} = 1, 2, \ldots, n_{ep}^{(1)}$ **do**

- $s_{v;e}^{(1)} = \mathrm{MLP}(X; \theta)$ ▷ MLP forward pass
- $\boxed{\mathcal{L}^{(1)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}}$ ▷ Eq. (1)
- Update θ w.r.t. $\frac{\partial \mathcal{L}^{(1)}}{\partial \theta}$ ▷ Gradient descent

return $s_{v;e}^{(1)} = \mathrm{MLP}(X; \theta)$ ▷ Trained scores



- Minimizing loss $\mathcal{L}^{(1)}$ → In each group e, its anchor $A(e)$ has a higher score compared to the other nodes in e

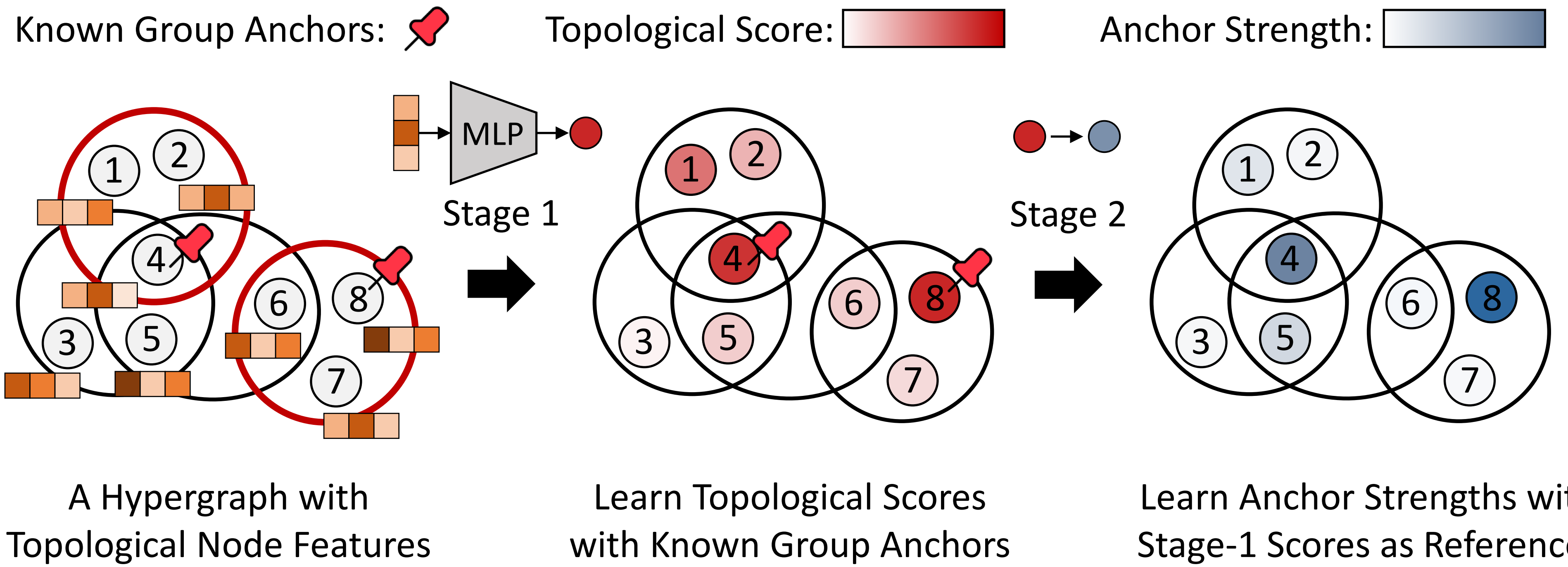
Proposed Method AnchorRadar: Stage 2

Observation 2: In real-world group interactions, whether a node is the group anchor or not is overall stable.

- After Stage 1, we have *topological scores* $s_{v;e}^{(1)}$'s
  - For each node v, its scores are defined *locally* within each group e
- Observation 2 tells us the *cross-group stability* of anchorship, and we also have the hypothesis that each node v has a *global anchor strength* shared across all groups involving v

Proposed Method ANCHORRADAR: Stage 2

- Learn a *global anchor strength* for each node v, so that
  - (1) The strengths **well explain the known anchors**
    - In each group, the anchor should have the highest strength
  - (2) The strengths **well align with the topological scores from Stage 1**



Proposed Method AnchorRadar: Stage 2

- Learn a *global anchor strength* $s_v^{(2)}$ for each node v, so that
  - (1) The strengths well explain the known anchors
    - In each group, the anchor should have the highest strength
  - (2) The strengths well align with the topological scores from Stage 1
- Minimizing $\mathcal{L}_1^{(2)}$ → In each group e, its *anchor* $A(e)$ *has a higher strength* compared to the other nodes in e

---

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s_{v;e}^{(1)}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n_{ep}^{(2)}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

1 $s_v^{(2)} \leftarrow 1, \forall v \in V$ ▷ Initialization

2 **for** $i_{ep} = 1, 2, \ldots, n_{ep}^{(2)}$ **do**

3 $\boxed{\mathcal{L}_1^{(2)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e)}^{(2)})}{\sum_{u \in e} \exp(s_u^{(2)})}}$ ▷ Eq. (2)

4 $\mathcal{L}_2^{(2)} = -\sum_{e \in E'} \frac{\exp(s_{A(e)}^{(2)})}{\sum_{u \in e} \exp(s_u^{(2)})} \cdot \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}$ ▷ Eq. (3)

5 Update each $s_v^{(2)}$ w.r.t. $\frac{\partial\left(\mathcal{L}_1^{(2)} + \alpha^{(2)} \mathcal{L}_2^{(2)}\right)}{\partial s_v^{(2)}}$ ▷ Gradient descent

6 $\hat{A}(e) = \arg\max_{v^* \in e} s_{v^*}^{(2)}, \forall e \in E$ ▷ Max anchor strength

7 $\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} \mathbb{1}[A(e)=v] + \sum_{e \in E \setminus E'} \mathbb{1}[\hat{A}(e)=v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

8 **return** $\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'$ ▷ Final prediction

Proposed Method AnchorRadar: Stage 2

- Learn a *global anchor strength* $s_v^{(2)}$ for each node v, so that
  - (1) The strengths well explain the known anchors
    - In each group, the anchor should have the highest strength
  - (2) The strengths well align with the topological scores from Stage 1
- Minimizing $\mathcal{L}_2^{(2)}$ → In each group e, the Stage-1 topological scores $s_{\cdot;e}^{(1)}$'s and the Stage-2 anchor strengths $s_{\cdot}^{(2)}$'s are *well-aligned*

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s_{v;e}^{(1)}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n_{ep}^{(2)}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

$s_v^{(2)} \leftarrow 1, \forall v \in V$ ▷ Initialization

for $i_{ep} = 1, 2, \ldots, n_{ep}^{(2)}$ **do**

- $\mathcal{L}_1^{(2)} = -\sum_{e \in E'} \log \frac{\exp(s_{A(e)}^{(2)})}{\sum_{u \in e} \exp(s_u^{(2)})}$ ▷ Eq. (2)
- $\boxed{\mathcal{L}_2^{(2)} = -\sum_{e \in E'} \frac{\exp(s_{A(e)}^{(2)})}{\sum_{u \in e} \exp(s_u^{(2)})} \cdot \frac{\exp(s_{A(e);e}^{(1)})}{\sum_{u \in e} \exp(s_{u;e}^{(1)})}}$ ▷ Eq. (3)
- Update each $s_v^{(2)}$ w.r.t. $\frac{\partial\left(\mathcal{L}_1^{(2)} + \alpha^{(2)} \mathcal{L}_2^{(2)}\right)}{\partial s_v^{(2)}}$ ▷ Gradient descent

$\hat{A}(e) = \arg\max_{v^* \in e} s_{v^*}^{(2)}, \forall e \in E$ ▷ Max anchor strength

$\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} 1[A(e)=v] + \sum_{e \in E \setminus E'} 1[\hat{A}(e)=v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

return $\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'$ ▷ Final prediction

Proposed Method AnchorRadar: Stage 2

- The final loss is a weighted sum of the two sub-losses $\mathcal{L}_1^{(2)}$ and $\mathcal{L}_2^{(2)}$
- Loss term coefficient $\alpha^{(2)}$ adjusts the emphasis between them
  - Using a higher $\alpha^{(2)}$
    → We emphasize $\mathcal{L}_2^{(2)}$ more
    → We emphasize the alignment between the two stages more

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s^{(1)}_{v;e}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n^{(2)}_{ep}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

```
s_v^(2) ← 1, ∀v ∈ V                                         ▷ Initialization
for i_ep = 1, 2, ..., n_ep^(2) do
    L_1^(2) = −Σ_{e∈E'} log [ exp(s^(2)_{A(e)}) / Σ_{u∈e} exp(s_u^(2)) ]      ▷ Eq. (2)
    L_2^(2) = −Σ_{e∈E'} [ exp(s^(2)_{A(e)}) / Σ_{u∈e} exp(s_u^(2)) ] · [ exp(s^(1)_{A(e);e}) / Σ_{u∈e} exp(s^(1)_{u;e}) ]   ▷ Eq. (3)
    Update each s_v^(2) w.r.t. ∂(L_1^(2) + α^(2) L_2^(2)) / ∂s_v^(2)    ▷ Gradient descent
Â(e) = arg max_{v*∈e} s^(2)_{v*}, ∀e ∈ E                    ▷ Max anchor strength
p̂_v = [ w^(2) Σ_{e∈E'} 1[A(e)=v] + Σ_{e∈E∖E'} 1[Â(e)=v] ] / d_v, ∀v ∈ V    ▷ Eq. (4)
return Ã(e) = arg max_{v*∈e} p̂_{v*}, ∀e ∈ E ∖ E'          ▷ Final prediction
```

Proposed Method AnchorRadar: Stage 2

- After training, we get the global anchor strength $s_v^{(2)}$ of each node v
- In each group e, $\hat{A}(e)$ is the node with the highest anchor strength

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s^{(1)}_{v;e}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n^{(2)}_{ep}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

$s_v^{(2)} \leftarrow 1, \forall v \in V$ ▷ Initialization

for $i_{ep} = 1, 2, \ldots, n^{(2)}_{ep}$ **do**

  $\mathcal{L}_1^{(2)} = -\sum_{e \in E'} \log \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s_u^{(2)})}$ ▷ Eq. (2)

  $\mathcal{L}_2^{(2)} = -\sum_{e \in E'} \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s_u^{(2)})} \cdot \frac{\exp(s^{(1)}_{A(e);e})}{\sum_{u \in e} \exp(s^{(1)}_{u;e})}$ ▷ Eq. (3)

  Update each $s_v^{(2)}$ w.r.t. $\frac{\partial\left(\mathcal{L}_1^{(2)} + \alpha^{(2)} \mathcal{L}_2^{(2)}\right)}{\partial s_v^{(2)}}$ ▷ Gradient descent

$\boxed{\hat{A}(e) = \arg\max_{v^* \in e} s^{(2)}_{v^*}, \forall e \in E}$ ▷ Max anchor strength

$\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} \mathbb{1}[A(e)=v] + \sum_{e \in E \setminus E'} \mathbb{1}[\hat{A}(e)=v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

return $\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'$ ▷ Final prediction

Proposed Method AnchorRadar: Stage 2

- In each group e, $\hat{A}(e)$ is the node with the highest anchor strength
- Then we do **global aggregation**: For each node v, we aggregate its anchorship information from all the groups involving v
  - If v is the known anchor $A(e)$ in a group e → it gets $w^{(2)}$ score
  - If v is the predicted anchor $\hat{A}(e)$ in a group e → it gets 1 score
- The global aggregation weight $w^{(2)}$ is used to give known information more credits than our predictions

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s^{(1)}_{v;e}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight **(6)** $n^{(2)}_{ep}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

$s^{(2)}_v \leftarrow 1, \forall v \in V$ ▷ Initialization

for $i_{ep} = 1, 2, \ldots, n^{(2)}_{ep}$ **do**

  $\mathcal{L}^{(2)}_1 = -\sum_{e \in E'} \log \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)}$ ▷ Eq. (2)

  $\mathcal{L}^{(2)}_2 = -\sum_{e \in E'} \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)} \cdot \frac{\exp(s^{(1)}_{A(e);e})}{\sum_{u \in e} \exp(s^{(1)}_{u;e})}$ ▷ Eq. (3)

  Update each $s^{(2)}_v$ w.r.t. $\frac{\partial\left(\mathcal{L}^{(2)}_1 + \alpha^{(2)} \mathcal{L}^{(2)}_2\right)}{\partial s^{(2)}_v}$ ▷ Gradient descent

$\hat{A}(e) = \arg\max_{v^* \in e} s^{(2)}_{v^*}, \forall e \in E$ ▷ Max anchor strength

$\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} 1[A(e)=v] + \sum_{e \in E \setminus E'} 1[\hat{A}(e)=v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

return $\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'$ ▷ Final prediction

Proposed Method AnchorRadar: Stage 2

- In each group e, the final prediction $\tilde{A}(e)$ is the node with the highest **globally aggregated score**

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s^{(1)}_{v;e}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n^{(2)}_{ep}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

$s^{(2)}_v \leftarrow 1, \forall v \in V$ ▷ Initialization

for $i_{ep} = 1, 2, \ldots, n^{(2)}_{ep}$ **do**

$\quad \mathcal{L}^{(2)}_1 = -\sum_{e \in E'} \log \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)}$ ▷ Eq. (2)

$\quad \mathcal{L}^{(2)}_2 = -\sum_{e \in E'} \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)} \cdot \frac{\exp(s^{(1)}_{A(e);e})}{\sum_{u \in e} \exp(s^{(1)}_{u;e})}$ ▷ Eq. (3)

$\quad$ Update each $s^{(2)}_v$ w.r.t. $\frac{\partial\left(\mathcal{L}^{(2)}_1 + \alpha^{(2)} \mathcal{L}^{(2)}_2\right)}{\partial s^{(2)}_v}$ ▷ Gradient descent

$\hat{A}(e) = \arg\max_{v^* \in e} s^{(2)}_{v^*}, \forall e \in E$ ▷ Max anchor strength

$\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} \mathbb{1}[A(e)=v] + \sum_{e \in E \setminus E'} \mathbb{1}[\hat{A}(e)=v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

return $\boxed{\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'}$ ▷ Final prediction

---

Proposed Method AnchorRadar: Stage 2

- **Intuition behind global aggregation:** It helps correct local errors, and thus increase the **robustness**
  - Can be understood as **majority vote**
- **Example:** The local error in the group {1,2,5} is corrected after global aggregation

Group e	Ground truth $A(e)$	Local pred. $\hat{A}(e)$	Final pred. $\tilde{A}(e)$
{1,2,3}	1	1	1
{1,2,4}	1	1	1
{1,2,5}	1	2	1

Algorithm 2: AnchorRadar: Stage 2

Input: **(1)** V and E: topology; **(2)** E' and $A(e), \forall e \in E'$: known group anchors; **(3)** $s^{(1)}_{v;e}, \forall v \in e \in E$: learned strengths from Stage 1; **(4)** $\alpha^{(2)}$: loss term coefficient; **(5)** $w^{(2)}$: global aggregation weight; **(6)** $n^{(2)}_{ep}$: number of optimization epochs

Output: $\tilde{A}(e), \forall e \in E \setminus E'$: predicted group anchors

$s^{(2)}_v \leftarrow 1, \forall v \in V$ ▷ Initialization

for $i_{ep} = 1, 2, \ldots, n^{(2)}_{ep}$ **do**

  $\mathcal{L}^{(2)}_1 = -\sum_{e \in E'} \log \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)}$ ▷ Eq. (2)

  $\mathcal{L}^{(2)}_2 = -\sum_{e \in E'} \frac{\exp(s^{(2)}_{A(e)})}{\sum_{u \in e} \exp(s^{(2)}_u)} \cdot \frac{\exp(s^{(1)}_{A(e);e})}{\sum_{u \in e} \exp(s^{(1)}_{u;e})}$ ▷ Eq. (3)

  Update each $s^{(2)}_v$ w.r.t. $\frac{\partial\left(\mathcal{L}^{(2)}_1 + \alpha^{(2)} \mathcal{L}^{(2)}_2\right)}{\partial s^{(2)}_v}$ ▷ Gradient descent

$\boxed{\hat{A}(e) = \arg\max_{v^* \in e} s^{(2)}_{v^*}, \forall e \in E}$ ▷ Max anchor strength

$\hat{p}_v = \frac{w^{(2)} \sum_{e \in E'} 1[A(e) = v] + \sum_{e \in E \setminus E'} 1[\hat{A}(e) = v]}{d_v}, \forall v \in V$ ▷ Eq. (4)

return $\boxed{\tilde{A}(e) = \arg\max_{v^* \in e} \hat{p}_{v^*}, \forall e \in E \setminus E'}$ ▷ Final prediction

Experimental Settings: Datasets

- We use 13 datasets from 5 different domains
- Train/Validation/Test = 7.5%/2.5%/90%

Domain	Dataset	Abbrev.	|V|	|E|	|E*|	Min. |e|	Max. |e|	Avg. |e|
Co-authorship (D_{co})	AMinerAuthor [21], [22]	coAA	1,712,433	2,037,605	1,454,250	1	115	2.55
	DBLP [22], [23]	coDB	108,476	91,260	81,601	2	36	3.52
	ScopusMultilayer [24]–[27]	coSM	1,673	937	842	1	27	3.09
Online Q&A (D_{qa})	StackOverflowBiology [22]	qaBI	15,418	26,290	23,242	1	12	2.08
	StackOverflowPhysics [22]	qaPH	80,434	194,575	169,274	1	40	2.38
	MathOverflow [24]	qaMA	410	154	154	2	57	4.27
	StackOverflow [24]	qaST	22,131	4,716	4,713	1	59	5.79
Email (D_{em})	EmailEnron [22]	emEN	21,251	101,124	34,916	2	883	11.53
	EmailEu [22], [28]	emEU	986	209,508	24,520	2	40	2.56
	Enron [24]	emER	110	9,603	1,169	2	29	2.47
Social network (D_{so})	Message [24]	soME	26,059	34,577	22,700	2	14	2.58
	Retweet [24]	soRE	30,073	88,148	49,828	2	2	2.00
Movie cast (D_{mo})	MovieLens [24], [29]	moML	73,155	43,058	42,497	1	5	4.70

*Data source: https://github.com/young917/EdgeDependentNodeLabel [22] and https://andrewmellor.co.uk/data/ [24].

Experimental Settings: Baselines

- Since we are the first to consider the problem of group anchor identification, **no immediate baselines exist**
- We adapt **existing methods originally proposed for a related problem**, edge-dependent node classification
- We have **9 baselines in total**:
  - WHATsNet, CoNHD-U, CoNHD-I, HNHN, HGNN, HCHA, HAT, UniGCN, HNN

[WHATsNet] Minyoung Choe et al. "Classification of Edge-Dependent Labels of Nodes in Hypergraphs." KDD'23

[CoNHD] Yijia Zheng et al. "Co-Representation Neural Hypergraph Diffusion for Edge-Dependent Node Classification." arXiv:2405.14286

[HNHN] Yihe Dong et al. "HNHN: Hypergraph Networks with Hyperedge Neurons." arXiv:2006.12278

[HGNN] Yifan Feng et al. " Hypergraph Neural Networks." AAAI'19

[HCHA] Song Bai et al. "Hypergraph Convolution and Hypergraph Attention." Pattern Recognition 110 (2021): 107637

[HAT] Hyunjin Hwang et al. "HyFER: A Framework for Making Hypergraph Learning Easy, Scalable and Benchmarkable." WWW-GLB'21

[UniGCN] Jing Huang and Jie Yang. "UniGNN: A Unified Framework for Graph and Hypergraph Neural Networks." IJCAI'21

[HNN] Ryan Aponte et al. "A Hypergraph Neural Network Framework for Learning Hyperedge-Dependent Node Embeddings." arXiv:2212.14077

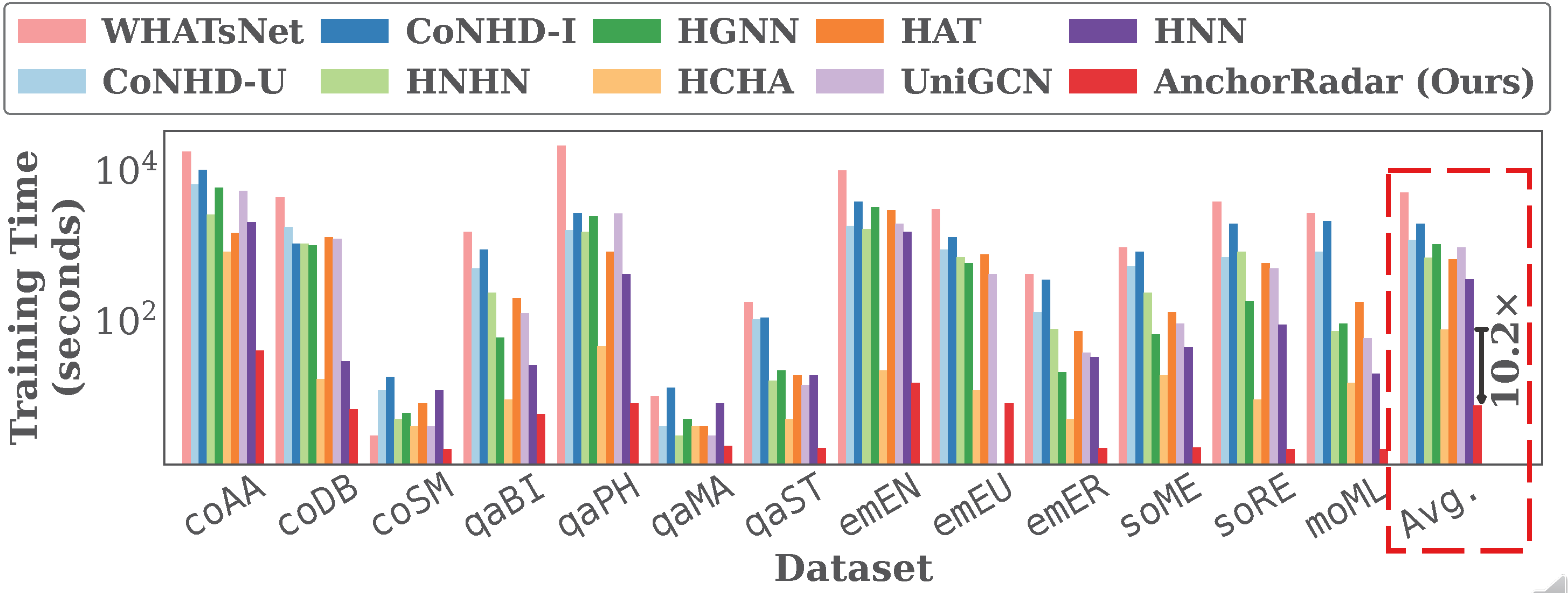
Results: ANCHORRADAR Achieves Higher Accuracy

- The proposed method ANCHORRADAR achieves the highest accuracy than all the baselines in most cases
- **Why?** Under label scarcity, the baselines that use deep neural networks and thus are heavily parameterized are *prone to overfitting*
- ANCHORRADAR's lightweight (MLP architecture), observation-driven design is more robust and alleviates this issue

Dataset	coAA (first)	coAA (last)	coDB (first)	coDB (last)	coSM (first)	coSM (last)	qaBI	qaPH	qaMA	qaST	emEN	emEU	emER	soME	soRE	moML
WHATsNet	45.2±0.2	45.8±0.3	42.5±0.3	45.4±0.2	34.3±2.0	39.8±1.8	85.6±0.4	88.1±0.1	35.8±1.1	31.2±0.2	50.8±3.4	51.0±0.3	66.6±3.1	**75.5±0.2**	97.4±0.3	41.4±0.3
CoNHD-U	42.7±1.3	42.2±2.0	41.2±0.1	42.7±0.3	31.4±1.9	37.9±2.1	78.7±0.5	76.0±1.0	29.0±4.2	25.4±3.8	44.0±4.3	**52.8±0.3**	65.3±2.2	74.3±0.3	96.8±0.6	42.4±0.5
CoNHD-I	44.5±0.8	44.7±0.3	41.4±0.5	43.5±0.7	29.8±2.6	39.4±1.9	79.3±0.4	77.3±0.2	29.8±6.1	26.6±1.3	45.1±3.8	52.4±0.4	64.6±1.2	74.6±0.4	97.5±0.6	42.7±0.3
HNHN	39.7±0.0	41.2±0.0	35.5±0.4	39.1±0.4	33.2±1.4	33.7±0.6	63.5±1.4	37.7±0.1	30.5±0.8	22.9±0.1	35.8±2.1	49.2±1.2	41.8±6.4	56.4±0.4	53.4±0.8	35.2±0.3
HGNN	44.1±0.0	45.9±0.1	41.9±0.1	44.6±0.3	33.1±0.3	38.1±0.6	81.7±0.3	74.9±0.8	28.9±1.0	30.4±0.5	40.1±0.7	49.3±0.2	42.0±0.9	62.1±2.1	84.6±3.4	37.9±0.5
HCHA	38.9±0.2	39.4±0.4	35.3±0.6	31.4±0.7	33.2±1.1	35.2±3.6	69.8±2.1	68.0±1.5	31.0±2.4	23.4±2.6	18.8±2.2	45.4±0.5	46.0±4.9	30.7±2.2	52.7±0.6	17.3±0.4
HAT	43.5±0.3	45.8±0.1	38.1±1.4	40.5±2.0	30.1±0.5	33.0±1.0	75.8±0.3	81.3±0.2	29.2±1.2	23.9±0.2	49.8±1.7	50.8±0.4	42.3±7.0	68.0±0.9	92.4±0.9	36.1±0.8
UniGCN	43.3±0.5	45.8±0.4	41.2±0.7	45.8±0.6	34.8±2.7	39.2±4.2	76.3±1.2	78.0±1.4	35.0±3.4	30.7±0.4	45.3±2.7	49.6±0.7	55.5±2.7	68.9±1.3	88.1±0.6	40.1±1.8
HNN	37.7±0.1	39.3±0.1	31.4±0.8	36.3±1.2	32.7±0.7	36.8±0.8	63.8±0.9	62.6±0.6	29.2±3.2	24.8±0.5	38.7±2.7	OOM	47.1±4.3	56.0±0.6	57.2±7.6	33.8±0.6
ANCHORRADAR	**49.7±0.1**	**50.6±0.0**	**46.5±0.2**	**49.9±0.1**	**40.9±0.7**	**48.1±1.3**	**87.4±0.2**	**88.7±0.1**	**40.6±5.3**	**36.6±0.4**	**53.6±2.4**	50.9±0.2	**67.8±2.6**	74.9±0.6	**97.8±0.3**	**45.0±0.3**

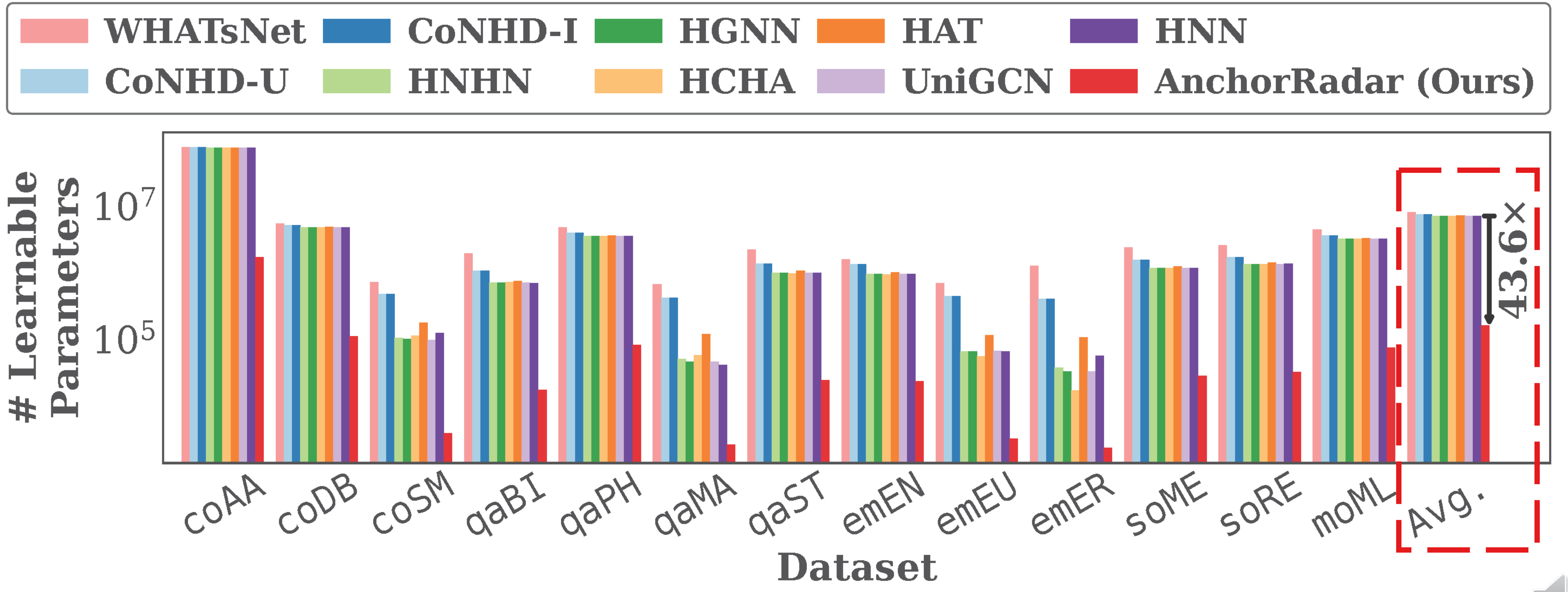
Results: AnchorRadar Uses Less Time and Parameters

- On average, the proposed method AnchorRadar uses **10.2× less training time** than the fastest baseline



Results: AnchorRadar Uses Less Time and Parameters

- On average, the proposed method AnchorRadar uses **43.6× fewer learnable parameters** than the most lightweight baseline



Results: Each Component in ANCHORRADAR is Helpful

- Variants of ANCHORRADAR excluding one component:
  - **Stage 1:** Excluding Stage 2, using the Stage-1 scores for final prediction
  - **Stage 2:** Excluding Stage 1, learning strengths without Stage 1's guidance
  - **No global aggregation:** Excluding the “majority vote” step
  - **No local features:** Excluding the group-specific topological features
- The full-fledged ANCHORRADAR outperforms all the variants, showing:
  - Every component positively contributes to its performance, and
  - The two stages create synergy

Dataset	coAA (first)	coAA (last)	coDB (first)	coDB (last)	coSM (first)	coSM (last)	qaBI	qaPH	qaMA	qaST	emEN	emEU	emER	soME	soRE	moML	Avg.
Stage 1	47.53	47.65	43.42	45.84	35.75	43.38	85.52	87.15	39.24	31.20	47.14	49.08	66.01	72.83	88.94	41.88	54.54
Stage 2	45.83	44.27	42.67	42.71	37.14	39.83	83.85	85.41	39.32	29.18	52.21	50.73	63.32	74.27	**97.82**	43.41	54.50
No global aggregation	49.50	**50.68**	45.83	49.90	40.12	48.13	86.47	87.78	**41.03**	35.96	53.07	**51.06**	67.63	73.67	96.03	**44.99**	57.62
No local features	49.64	50.54	**46.63**	49.86	40.53	**48.32**	87.26	88.69	40.08	36.48	**54.24**	50.87	66.96	**74.91**	**97.82**	43.32	57.89
ANCHORRADAR	**49.68**	50.60	46.55	**49.95**	**40.92**	48.08	**87.41**	**88.74**	40.59	**36.57**	53.55	50.89	**67.77**	74.87	**97.82**	**44.99**	**58.06**

Results: AnchorRadar is Helpful for Downstream Task

- **Task:** Group-interaction prediction
  - Specifically, distinguish real and fake group interactions
- **Backbone:** VilLain (a self-supervised method on hypergraphs)
  - VilLain obtains group (hyperedge) embeddings from topology
- We include group-level statistics of anchor strengths (e.g., mean and standard deviation) to enrich the group embeddings from VilLain
- The additional information from AnchorRadar further helps VilLain to better distinguish real and fake group interactions

Dataset	coDB (first)	coDB (last)	coSM (first)	coSM (last)	qaBI	qaPH	qaST	emEN	emEU	emER	soME	soRE	moML	Avg.
Original VilLain	89.71	89.71	91.40	91.40	71.68	74.58	**76.69**	89.65	87.22	87.45	96.76	94.39	95.56	87.40
+ Anchor Strengths	**93.11**	**93.41**	**91.94**	**92.47**	**80.45**	**85.64**	74.06	**97.57**	**92.01**	**90.73**	**97.87**	**95.82**	**96.52**	**90.89**

[VilLain] Geon Lee et al. "VilLain: Self-Supervised Learning on Homogeneous Hypergraphs without Features via Virtual Label Propagation." WWW'24

Conclusion

In this work, we…

- **Proposed New Concept and Problem:** Introduced the concept of group anchors, and the novel and practical problem of identifying them
- **Made Key Observations:** Grounded our work in real-world data, showing empirical patterns of anchors in real-world group interactions
- **Developed Effective Algorithm:** Proposed ANCHORRADAR, an intuitive, lightweight, and observation-driven method
- **Ran Extensive Experiments:** Demonstrated that ANCHORRADAR is more accurate, faster, and lighter than baselines

Appendix, Code, and Datasets: bit.ly/anchor_rader_ICDM25