KAIST AI
Kim Jaechul Graduate School

Birmingham 2023
CIKM

Robust Graph Clustering via Meta Weighting for Noisy Graphs

Hyeonsoo Jo

Fanchen Bu

Kijung Shin

Robust Graph Clustering via Meta Weighting for Noisy Graphs

Introduction

Preliminaries

Method

Experiment

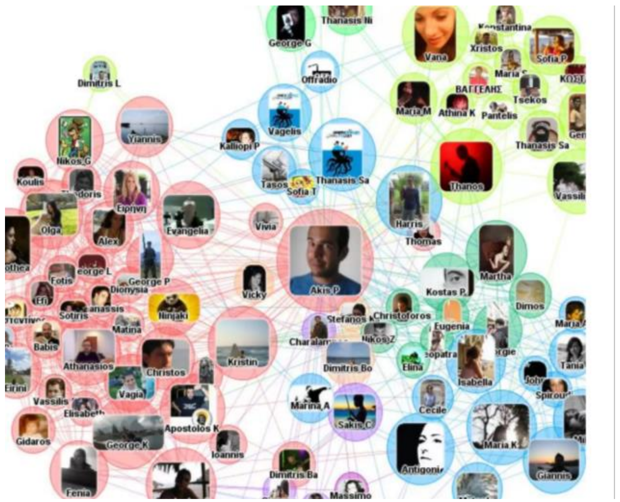
Conclusions

Robust Graph Clustering via Meta Weighting for Noisy Graphs

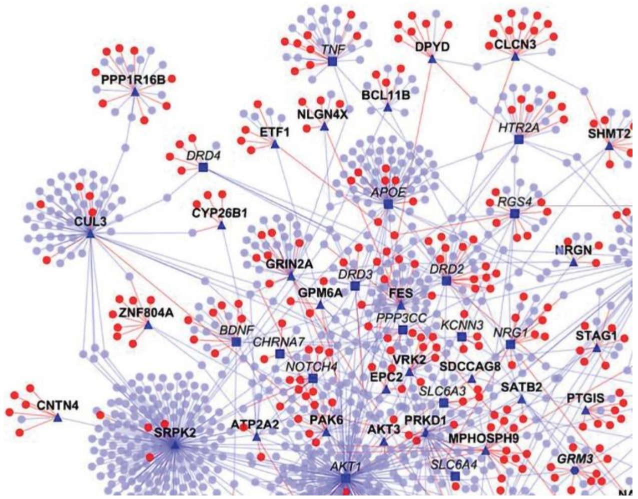
- **Introduction**
- Preliminaries
- Method
- Experiment
- Conclusions

Clusters in Graphs

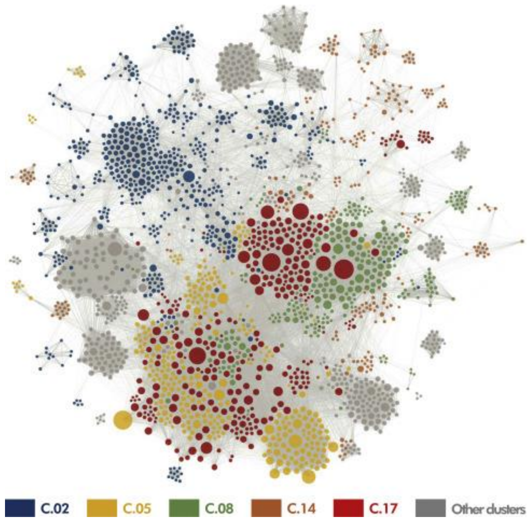
Finding clusters in real-world graphs **have useful implications.**



Social Groups
in friendship networks



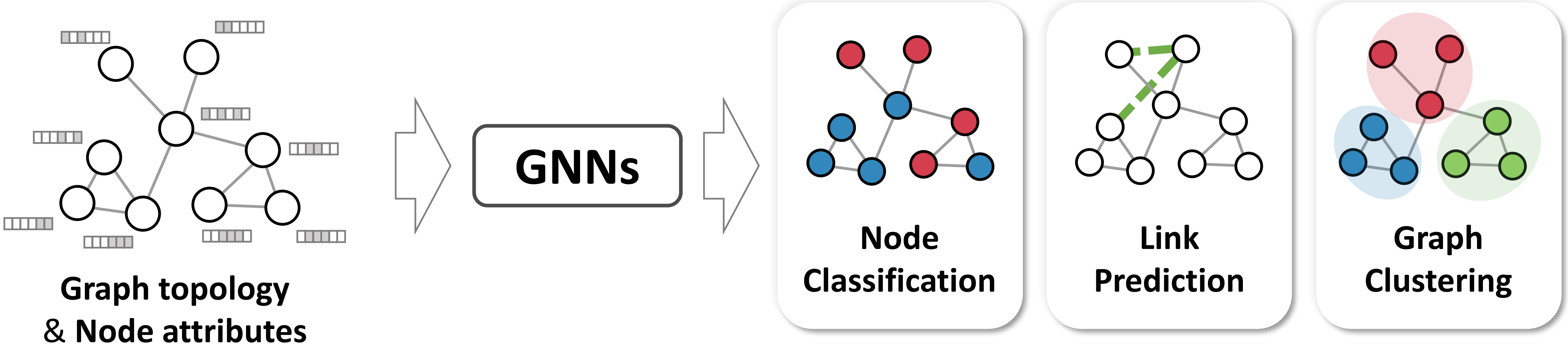
Functional modules
in Protein-interaction networks



Papers on the same topic
in citation networks

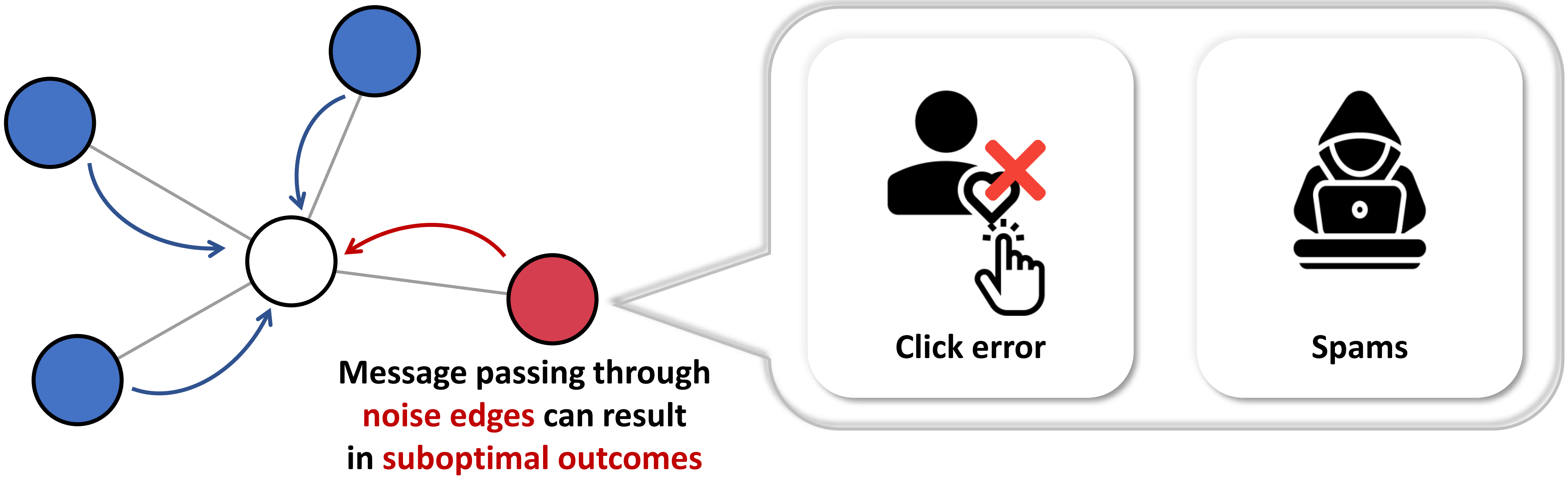
Graph Neural Networks (GNNs)

- GNNs achieve **remarkable performance** in **various tasks**.
- Several GNN-based approaches have been **developed also for graph clustering**.



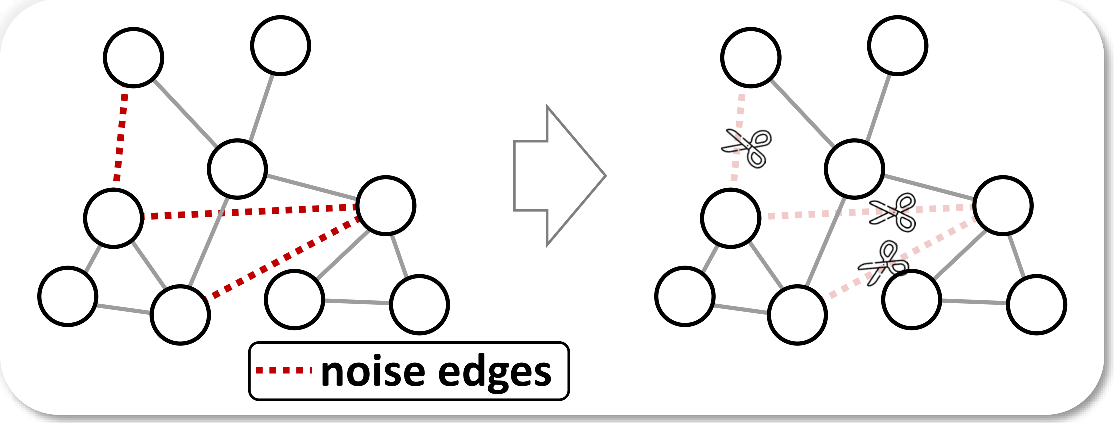
GNNs are Vulnerable to Noise Edges

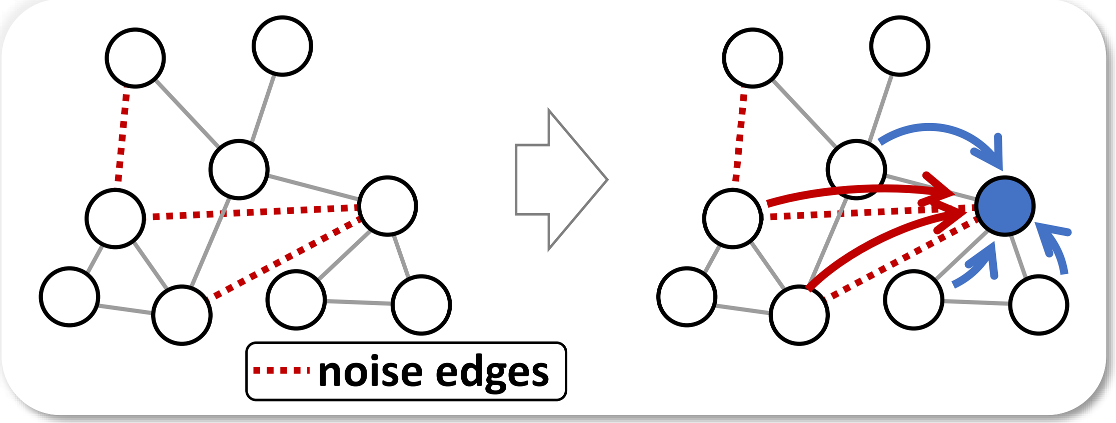
- GNN-based approaches in general are **vulnerable to noise edges** in graphs.
- Real-world graphs are often **contaminated by noise edges** (click error, spams, etc.)



Robust GNNs: Overview

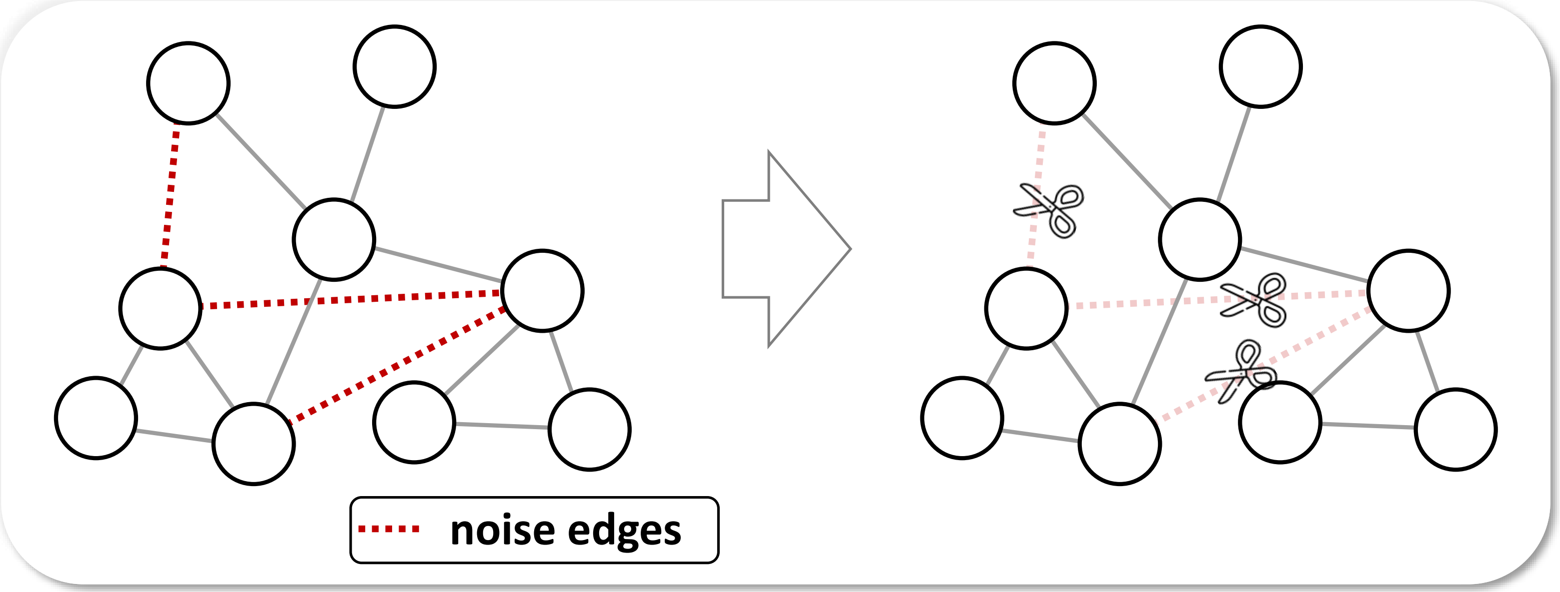
- Several methods have been developed for **robustness of GNNs**.
- They are categorized into **edge filtering** and **adjusted message-passing**.





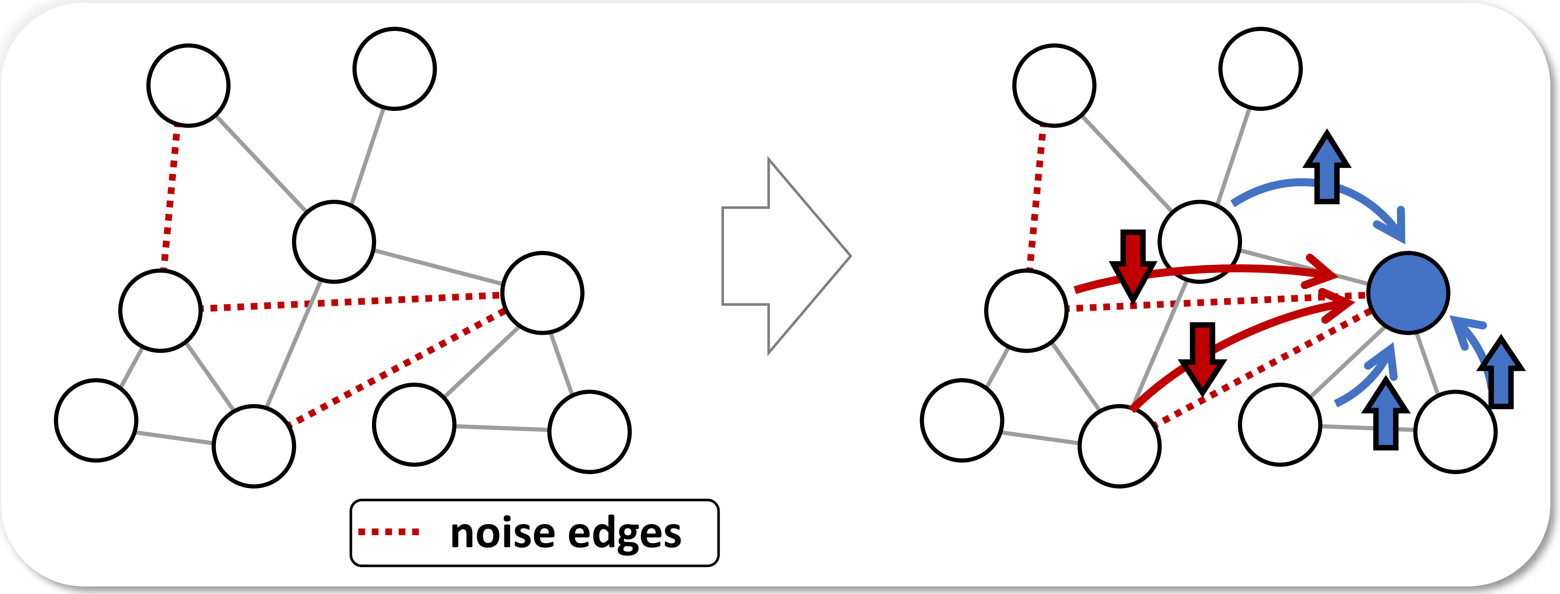
Robust GNNs: Edge Filtering

- Filtering edges based on **graph topology** or **node attribute-based similarity**.
- Learning a denoised adjacency matrix through **end-to-end learning with constraints** (e.g., ℓ_1 *norm* for sparsity, *nuclear norm* for low-rankness).



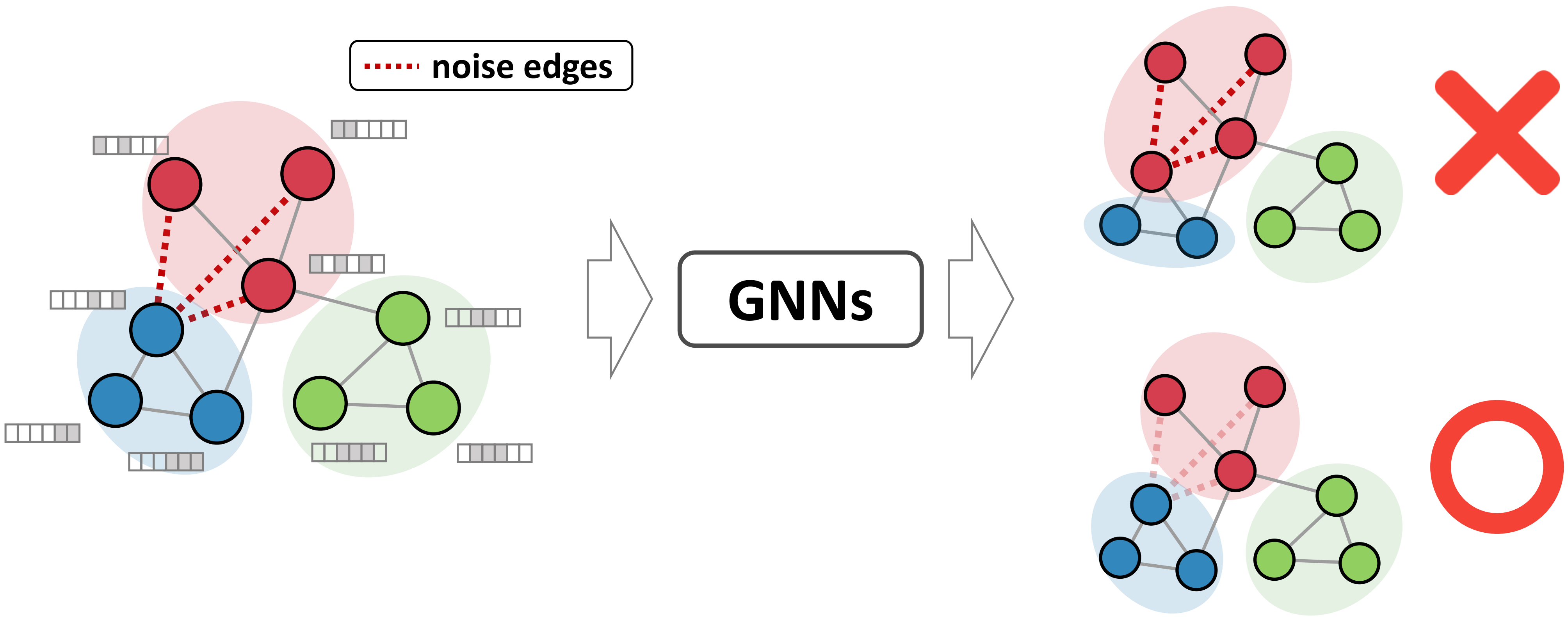
Robust GNNs: Adjust Message-Passing

- Adjust message-passing schemes to **reduce messages passed through noise edges**.
- It relies on **supervision** from **node-label** or **graph-label,** which are not contaminated by noise edges.



Robust Graph Clustering

Q How to improve the robustness of GNN-based graph clustering?



Challenges in Robust Graph Clustering

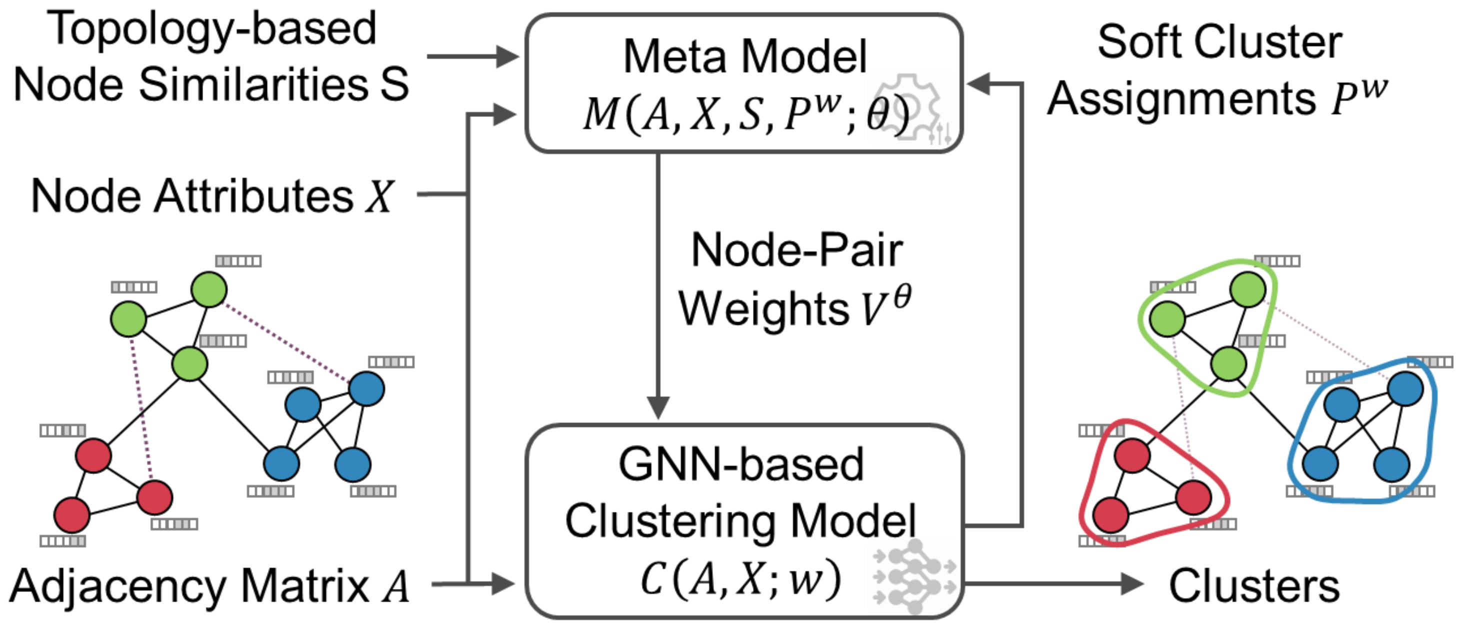
1. Lack of Supervision

- Existing robust GNNs rely on **supervision** from **node-labels** or **graph-labels**, which however, is **not given in graph clustering**.

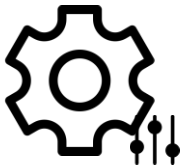
2. Contaminated Objective Function

- The **objective function** of graph clustering is **based on graph topology**, and thus it is **contaminated by noise edges**.

MetaGC (Meta-weighting based Graph Clustering)



Decomposable
clustering loss function



Meta weighting
adaptively adjust
the weight of each pair

Our Contributions

Observations

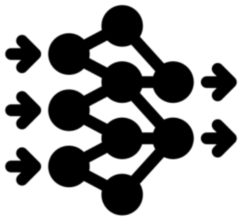
GNN-based graph clustering is **vulnerable to noise edges**.

Theory

Define a theoretically **suitable family of clustering loss functions** for GNN-based clustering.

Methodology

Develop a **robust graph clustering** method based on **Meta-Weighting.**



Experiments

Extensive experiments on **5 real-world graphs under 3 levels of noise**.

Robust Graph Clustering via Meta Weighting for Noisy Graphs

Introduction

Preliminaries

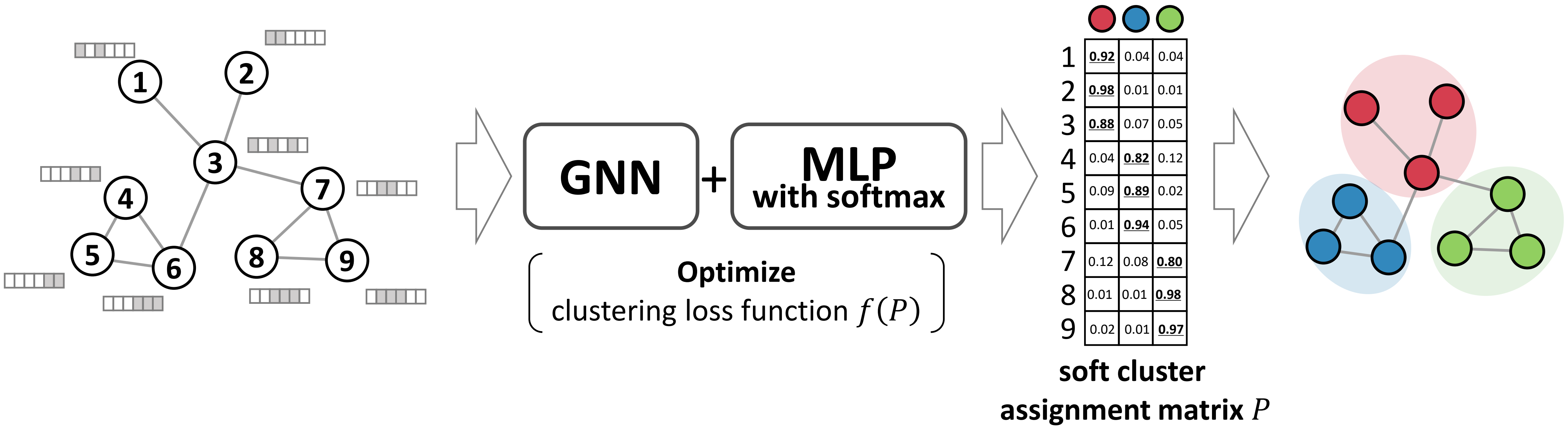
Method

Experiment

Conclusions

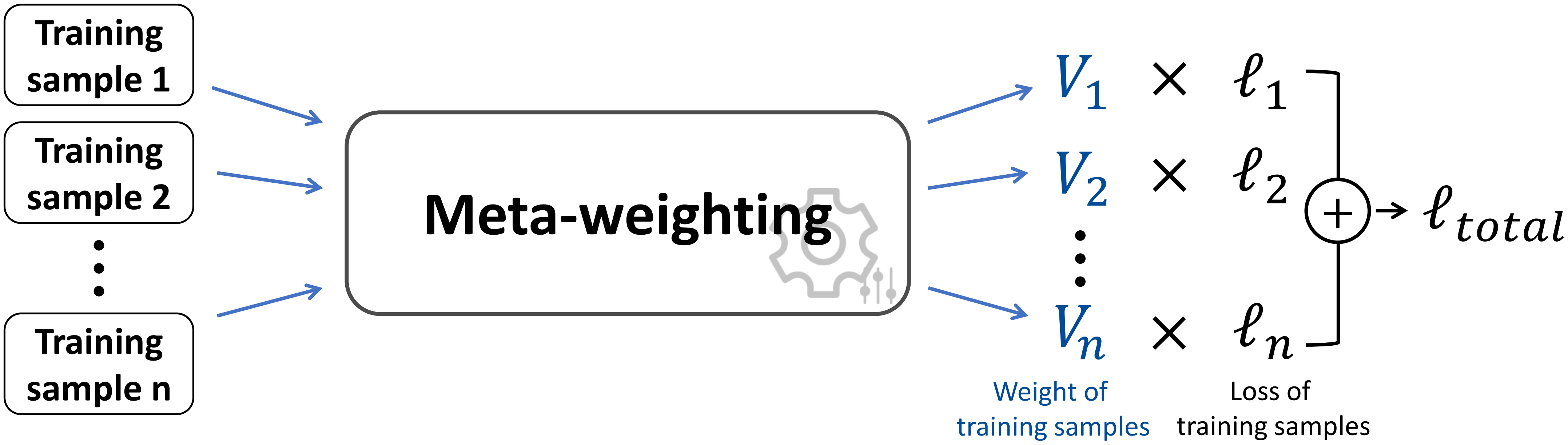
GNN-based Graph Clustering Methods

- Graph clustering GNNs use a **GNN** followed by a **MLP** and **softmax**.
- Their output is a soft cluster assignment matrix P.
- The objective function is a **continuous relaxation** of a **clustering loss function**.



Meta-Weighting

- Meta-weighting is a meta-learning-based method that **learns the weights of training samples** while **minimizing an objective function**.
- Meta-weighting has been successful for classification and recommendation.



Robust Graph Clustering via Meta Weighting for Noisy Graphs

Introduction

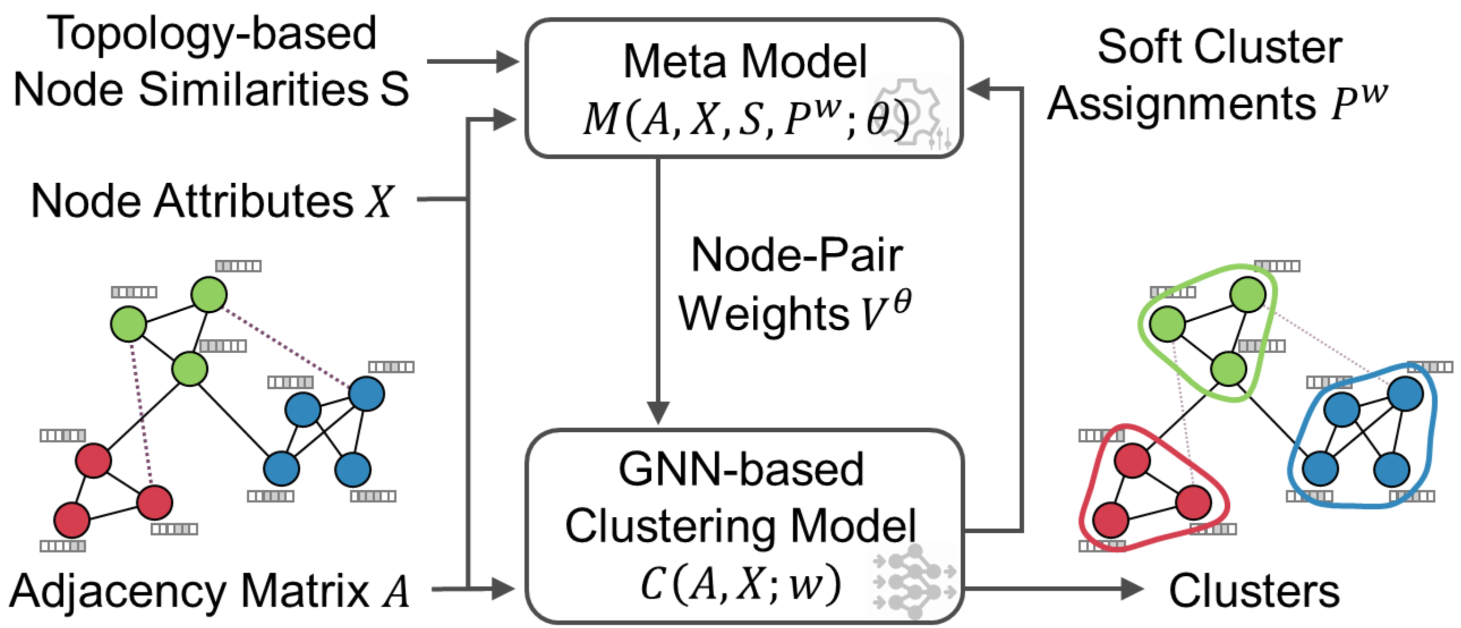
Preliminaries

Method

Experiment

Conclusions

Proposed Method: MetaGC



Decomposable
clustering loss function

Meta weighting
adaptively adjust
the weight of each pair

Decomposable Clustering Loss Function $f(P)$

- Can be **decomposed** into a summation of **loss** on **each node pairs**.
- **The loss on each pair** is a **weighted dot product** of their assignment vectors.

[Definition] ***Decomposable clustering loss functions***
Given $G = (V, E)$ with $|V| = N$, $\mathrm{k} \in \mathbb{N}$, and a soft cluster assignment matrix $P \in \mathcal{P}$, a clustering loss function $P \in \mathcal{P}$, a clustering loss function $f: \mathcal{P} \rightarrow \mathbb{R}$ ***decomposable***, if there exist constant $c_{ij} = c_{ij}(G), \forall i, j \in [N]$ s.t.

$$f(P) = \sum_{\substack{(i,j)\in \\ all\ node\ pairs}} c_{ij} P_i \cdot P_j$$

Decomposable Clustering Loss Function $f(P)$

- Decomposable clustering loss function is **suitable** for **continuous relaxation**.
- **[Theoretical guarantee]**
  optimal soft clustering assignments = ***optimal deterministic clustering assignments***
  (for details see *Sec 3.3*).
- We use a **modularity-based objective function**.

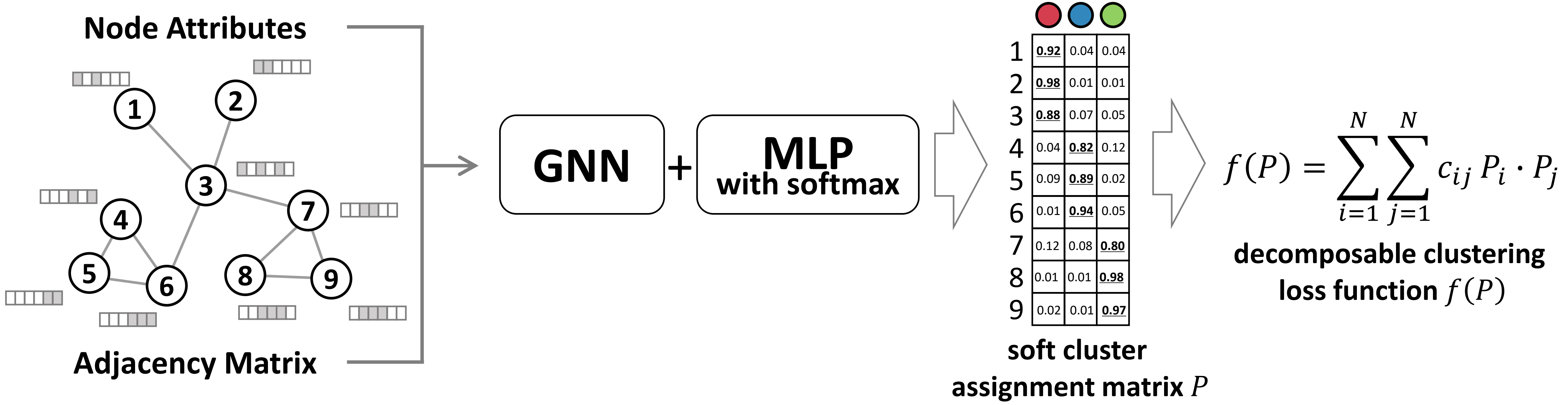
[Example] ***Modularity-based loss function***

Modularity-based clustering loss function is a representative objective for clustering.

$$f(P) = \sum_{\substack{(i,j)\in \\ \text{all node pairs}}} \underbrace{\frac{1}{2|E|}\left(A_{ij} - \frac{d_i d_j}{2|E|}\right)}_{= c_{ij}} P_i \cdot P_j$$

GNN-based Clustering Model C

- GNN-based clustering model C yields a **soft cluster assignment matrix P**.
- **Input of C**: Adjacency matrix, Node Attributes
- **Output of C**: Soft cluster assignment matrix P

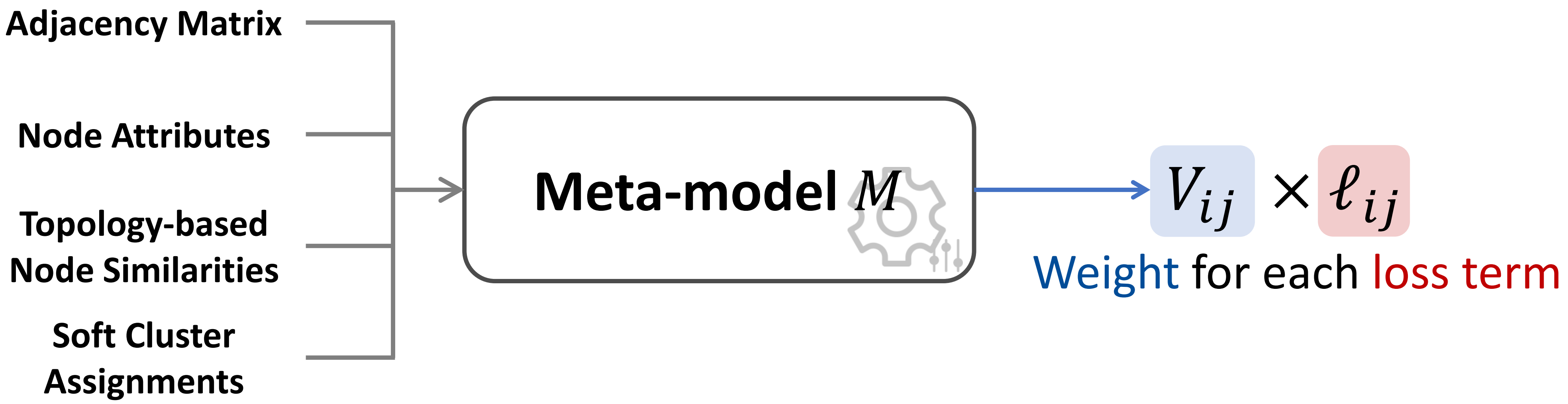


Meta-model M

- Meta-model M yields **a node-pair weight matrix V**, used to weight the loss function.
- **Input of M**: Adjacency matrix, Node Attributes,

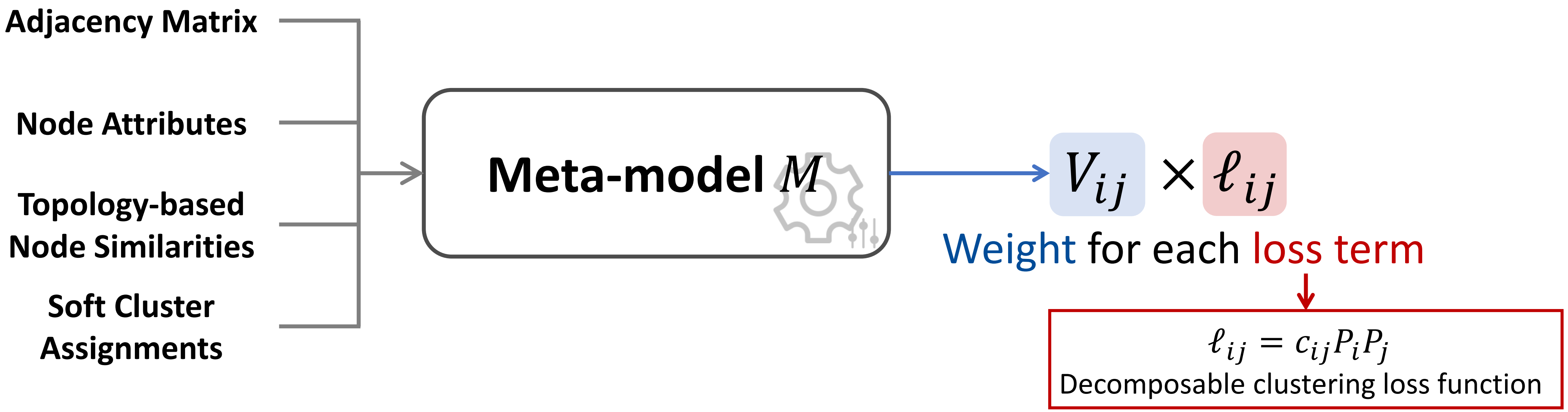
  Topology-based Node Similarities, Soft Cluster Assignments

  e.g., Common neighbors, Adamic-Adar index etc. The output of the clustering model C.
- **Output of M**: Soft cluster assignment matrix P



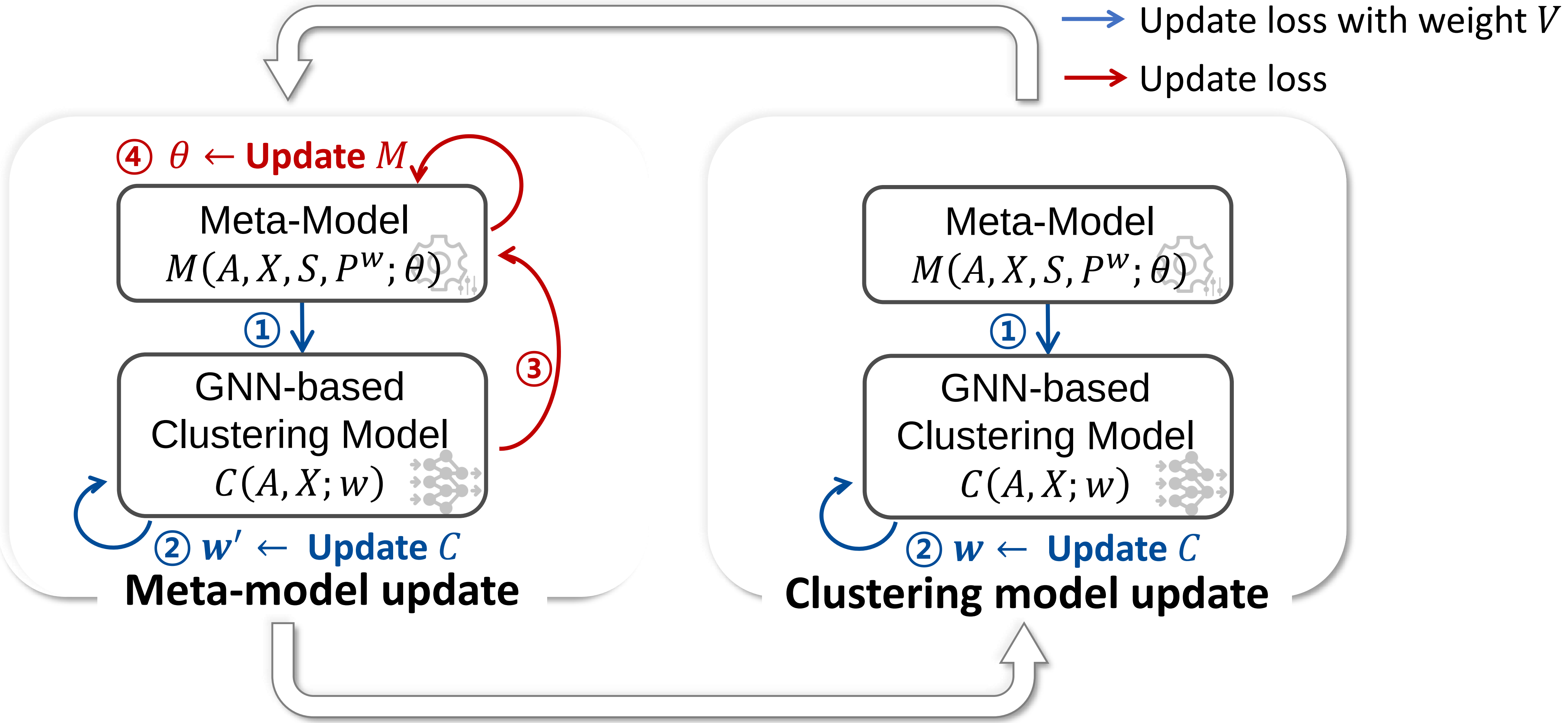
Meta-model M

- Meta-model M yields **a node-pair weight matrix V**, used to weight the loss function.
- **Input of M**: Adjacency matrix, Node Attributes,
  Topology-based Node Similarities, Soft Cluster Assignments
  → e.g., Common neighbors, Adamic-Adar index etc. → The output of the clustering model C.
- **Output of M**: Soft cluster assignment matrix P



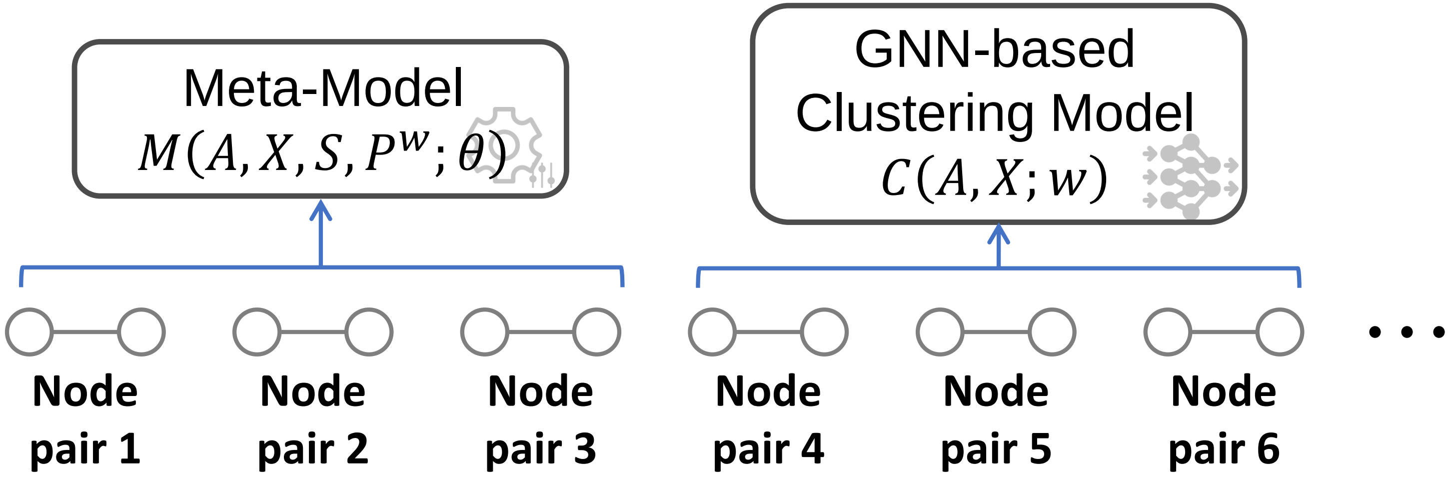
Overall Training Procedure

- We follow **the standard training procedure** for **meta-weighting**



Overall Training Procedure (Clean Validation)

- **Conventional Meta-Weighting**
  - Meta-weighting methods assume a **clean validation set** to **train a meta-model** .
  - In noisy graph, the clean parts are **typically unknown**.
- **Our Hypothesis: Clean validation set is not "necessary"**
  - Using **distinct batches** for training a **meta-model** and a **clustering model**.
  - Our empirical results show that it is still possible to train a meta-model meaningfully.



Robust Graph Clustering via Meta Weighting for Noisy Graphs

Introduction

Preliminaries

Method

Experiment

Conclusions

Experimental Questions

[Q1] Robustness & Accuracy

Is MetaGC more **robust** and **accurate** than the competitors on noisy graphs?

[Q2] Effectiveness of Meta-Weighting

Does the meta-model in MetaGC **properly adjust** the weights of loss terms?

[Q3] Ablation Study

Does each component of MetaGC **contribute** to **performance improvement**?

Experimental Setups (Cont.)

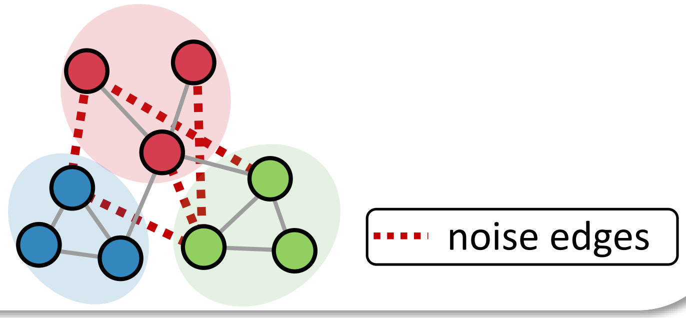
Dataset

- We use **5 real–world dataset**.
- 4 citation graphs and 1 co-purchase graph.

Name	# Nodes	# Edges	# Attributes	# Classes
Cora	2,708	5,278	1,433	7
Cora-ML	2,995	8,158	2,879	7
Citeseer	3,327	4,552	3,703	6
Amazon-Photo	7,535	119,081	745	8
Pubmed	19,717	44,324	500	3

Noisy-graph Generation

- We add noise edges **randomly**, selecting those with **endpoints in different classes**.
- ***Noise levels I, II, and III:*** **30%**, **60%**, and **90%** noise-to-existing edge ratios, respectively.



Experimental Setups (Cont.)

Competitors

- We compare MetaGC with **13 competitors**.
  - Four **node embedding based methods**
  - Three **GNN-based graph clustering methods**
  - Six **graph denoising methods**

Evaluation Metrics

- We use three evaluation metrics,
  - **F1 Score**
  - **Normalize Mutual Information (NMI)**
  - **Modularity**

(Q1) Robustness & Accuracy

- MetaGC gives the **best overall clustering performance** under each noise lebel.

Noise Level	I			II			III			AR
Metric	F1 Score	NMI	Modularity	F1 Score	NMI	Modularity	F1 Score	NMI	Modularity	
DeepWalk	0.405±0.048	0.465±0.008	0.689±0.006	0.297±0.022	0.389±0.010	0.659±0.007	0.256±0.014	0.352±0.012	0.641±0.007	6.0***
Node2Vec	0.410±0.043	0.464±0.006	0.690±0.005	0.296±0.023	0.389±0.008	0.660±0.004	0.261±0.017	0.359±0.011	0.642±0.007	5.4***
DGI	0.230±0.010	0.287±0.003	0.151±0.007	0.198±0.009	0.239±0.004	0.141±0.010	0.183±0.006	0.203±0.013	0.122±0.013	9.7***
GMI	0.099±0.004	0.021±0.001	-0.003±0.001	0.103±0.005	0.025±0.001	-0.002±0.001	0.109±0.006	0.030±0.001	-0.002±0.001	11.7***
MinCutPool	0.464±0.000	0.000±0.000	0.000±0.000	0.464±0.000	0.000±0.000	0.000±0.000	0.464±0.000	0.000±0.000	0.000±0.000	9.4***
DMoN	0.556±0.049	0.533±0.041	0.609±0.036	0.528±0.028	0.494±0.025	0.599±0.023	0.470±0.033	0.425±0.036	0.531±0.050	3.3***
GCC	0.538±0.022	0.501±0.039	0.619±0.034	0.469±0.007	0.377±0.019	0.540±0.027	0.459±0.006	0.353±0.018	0.526±0.024	5.9***
GCN-Jaccard	0.557±0.049	0.533±0.040	0.610±0.036	0.525±0.034	0.493±0.028	0.597±0.032	0.473±0.034	0.431±0.038	0.538±0.052	3.1***
GCN-SVD	0.390±0.004	0.365±0.009	0.497±0.002	0.408±0.005	0.379±0.004	0.506±0.006	0.403±0.005	0.374±0.017	0.507±0.011	7.4***
GDC	0.514±0.073	0.502±0.054	0.572±0.043	0.474±0.057	0.447±0.052	0.547±0.059	0.463±0.033	0.418±0.031	0.532±0.050	4.9***
ProGNN	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	N.A.
PTDNet	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	N.A.
FGC	0.377±0.000	0.071±0.001	0.145±0.003	0.366±0.000	0.055±0.000	0.103±0.001	0.362±0.000	0.048±0.000	0.084±0.001	9.6***
MetaGC	0.562±0.015	0.566±0.017	0.675±0.008	0.528±0.020	0.520±0.013	0.664±0.007	0.508±0.014	0.498±0.009	0.658±0.006	1.2

(a) Amazon-Photo

(Q1) Robustness & Accuracy

Noise Level	I			II			III			AR
Metric	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	
DeepWalk	0.300±0.024	0.243±0.010	0.680±0.009	0.216±0.010	0.155±0.006	0.593±0.011	0.169±0.014	0.111±0.009	0.528±0.008	8.3***
Node2Vec	0.292±0.028	0.247±0.015	0.684±0.009	0.210±0.016	0.154±0.010	0.594±0.009	0.170±0.009	0.111±0.011	0.528±0.009	8.3***
DGI	0.351±0.040	0.415±0.011	0.619±0.015	0.294±0.027	0.330±0.012	0.547±0.017	0.248±0.018	0.240±0.017	0.412±0.033	6.1***
GMI	0.277±0.023	0.319±0.008	0.576±0.010	0.226±0.016	0.229±0.005	0.496±0.007	0.152±0.016	0.145±0.012	0.391±0.020	9.7***
MinCutPool	0.265±0.035	0.222±0.023	0.614±0.012	0.217±0.027	0.147±0.019	0.556±0.012	0.219±0.097	0.086±0.039	0.436±0.172	10.1***
DMoN	0.400±0.023	0.343±0.015	0.661±0.012	0.355±0.023	0.280±0.013	0.620±0.013	0.326±0.016	0.231±0.016	0.576±0.012	4.6***
GCC	0.375±0.017	0.230±0.013	0.486±0.011	0.364±0.023	0.114±0.014	0.312±0.053	0.364±0.041	0.076±0.016	0.252±0.073	9.6***
GCN-Jaccard	0.415±0.022	0.364±0.017	0.661±0.014	0.369±0.030	0.310±0.014	0.627±0.013	0.348±0.030	0.276±0.017	0.602±0.016	2.7**
GCN-SVD	0.313±0.025	0.207±0.019	0.487±0.022	0.291±0.031	0.172±0.023	0.468±0.016	0.288±0.024	0.156±0.017	0.458±0.018	8.9***
GDC	0.298±0.030	0.218±0.021	0.577±0.020	0.266±0.027	0.183±0.017	0.555±0.011	0.269±0.010	0.175±0.016	[illegible]	8.1***
ProGNN	0.405±0.023	0.348±0.015	0.631±0.015	0.370±0.022	0.296±0.011	0.590±0.016	0.341±0.018	0.248±0.017	[illegible]	[illegible]
PTDNet	0.198±0.014	0.033±0.010	0.300±0.011	0.186±0.010	0.031±0.005	0.279±0.007	0.209±0.018	[illegible]	[illegible]	13.6***
FGC	0.388±0.005	0.145±0.005	0.337±0.006	0.374±0.005	0.123±0.006	0.314±0.006	[illegible]	[illegible]	[illegible]	8.7***
MetaGC	0.413±0.030	0.379±0.027	0.696±0.010	0.372±0.028	0.320±0.023	0.660±0.015	[illegible]	0.282±0.021	0.628±0.018	1.7

(b) Cora

Noise Level	I			II			III			AR
Metric	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	
DeepWalk	0.128±0.004	0.089±0.003	0.650±0.004	0.103±0.004	0.053±0.003	0.586±0.005	0.086±0.004	0.037±0.002	0.545±0.003	7.3***
Node2Vec	0.127±0.004	0.089±0.003	0.650±0.003	0.101±0.005	0.053±0.002	0.587±0.005	0.085±0.004	0.037±0.003	0.545±0.005	7.6***
DGI	0.199±0.019	0.108±0.003	0.169±0.005	0.160±0.007	0.064±0.002	0.153±0.007	0.130±0.006	0.043±0.003	0.160±0.007	9.2***
GMI	0.159±0.009	0.117±0.002	0.154±0.003	0.115±0.010	0.072±0.001	0.135±0.002	0.116±0.006	0.049±0.001	0.115±0.006	9.3***
MinCutPool	0.380±0.005	0.131±0.005	0.512±0.010	0.345±0.017	0.097±0.009	0.487±0.011	0.325±0.020	0.079±0.011	0.472±0.016	5.9***
DMoN	0.406±0.005	0.161±0.004	0.542±0.001	0.377±0.008	0.125±0.011	0.518±0.003	0.346±0.019	0.090±0.023	0.497±0.010	3.7***
GCC	[illegible]	[illegible]	0.313±0.005	0.459±0.004	0.039±0.001	0.421±0.004	0.505±0.053	0.019±0.010	0.276±0.149	7.4***
GCN-Jaccard	[illegible]	0.163±0.005	0.542±0.001	0.377±0.008	0.126±0.011	0.518±0.003	0.346±0.020	0.090±0.023	0.498±0.010	3.2***
GCN-SVD	[illegible]	0.094±0.022	0.379±0.006	0.351±0.038	0.076±0.014	0.372±0.006	0.332±0.035	0.060±0.014	0.365±0.005	7.1***
GDC	0.360±0.022	0.113±0.013	0.481±0.010	0.351±0.006	0.097±0.005	0.477±0.004	0.327±0.023	0.071±0.022	0.475±0.011	6.2***
ProGNN	O.O.T.	O.O.T.	O.O.T.	[illegible]	O.O.T.	O.O.T.	O.O.T.	O.O.T.	O.O.T.	N.A.
PTDNet	O.O.M.	O.O.M.	O.O.M.	[illegible]	O.O.M.	O.O.M.	O.O.M.	O.O.M.	O.O.M.	N.A.
FGC	0.598±0.000	0.000±0.000	[illegible]	[illegible]	0.059±0.000	0.261±0.000	0.576±0.000	0.044±0.000	0.218±0.000	7.1***
MetaGC	[illegible]	0.175±0.010	[illegible]	0.396±0.004	0.160±0.004	0.523±0.001	0.380±0.003	0.141±0.004	0.513±0.001	2.6

(d) Pubmed

Noise Level	I			II			III			AR
Metric	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	
DeepWalk	0.375±0.009	0.276±0.013	0.636±0.016	0.314±0.010	0.201±0.012	0.581±0.009	0.250±0.018	[illegible]	[illegible]	6.4***
Node2Vec	0.377±0.007	0.282±0.006	0.644±0.005	0.308±0.010	0.199±0.011	0.584±0.006	[illegible]	[illegible]	[illegible]	5.8***
DGI	0.379±0.063	0.295±0.037	0.340±0.045	0.242±0.009	0.131±0.015	[illegible]	[illegible]	0.063±0.017	0.083±0.027	9.8***
GMI	0.366±0.018	0.268±0.011	0.395±0.013	0.259±0.012	0.144±0.007	[illegible]	[illegible]	0.062±0.005	0.115±0.008	10.1***
MinCutPool	0.271±0.026	0.200±0.019	0.592±0.018	0.278±0.103	0.105±0.054	[illegible]	0.437±0.067	0.012±0.026	0.035±0.124	9.3***
DMoN	0.340±0.026	0.289±0.025	0.661±0.016	0.314±0.020	0.237±0.023	0.630±0.016	0.291±0.018	0.204±0.019	0.600±0.016	[illegible]
GCC	0.461±0.022	0.299±0.024	0.441±0.041	0.415±0.014	0.165±0.049	0.306±0.086	0.379±0.030	0.105±0.036	[illegible]	[illegible]
GCN-Jaccard	0.358±0.033	0.283±0.033	0.600±0.015	0.323±0.025	0.232±0.025	0.569±0.011	0.295±0.020	[illegible]	[illegible]	5.4***
GCN-SVD	0.275±0.022	0.165±0.024	0.403±0.029	0.247±0.017	0.142±0.011	0.365±0.015	0.261±0.016	[illegible]	0.338±0.016	9.4***
GDC	0.267±0.019	0.159±0.016	0.475±0.019	0.230±0.020	0.102±0.012	0.366±0.013	0.190±0.024	[illegible]	0.285±0.009	11.0***
ProGNN	0.345±0.026	0.297±0.025	0.662±0.016	0.319±0.020	0.244±0.023	0.631±0.015	0.297±0.018	0.212±0.020	0.603±0.015	3.6***
PTDNet	0.235±0.045	0.058±0.007	0.288±0.007	0.216±0.055	0.046±0.006	0.249±0.012	0.271±0.041	0.044±0.006	0.244±0.013	11.9***
FGC	0.395±0.013	0.035±0.011	0.111±0.029	0.415±0.002	0.017±0.003	0.059±0.005	0.424±0.003	0.010±0.001	0.038±0.006	9.8***
MetaGC	0.380±0.034	0.337±0.024	0.683±0.014	0.333±0.026	0.282±0.015	0.656±0.015	0.319±0.021	0.255±0.014	0.639±0.015	1.8

(c) Cora-ML

Noise Level	I			II			III			AR
Metric	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	F1 Score	NMI	Moduarity	
DeepWalk	0.177±0.013	0.083±0.005	[illegible]	0.136±0.012	0.054±0.007	0.656±0.010	0.112±0.014	0.044±0.004	0.596±0.007	9.7***
Node2Vec	0.180±0.011	[illegible]	[illegible]	0.146±0.013	0.057±0.004	0.661±0.007	0.116±0.011	0.041±0.005	0.596±0.011	9.2***
DGI	[illegible]	[illegible]	[illegible]	0.204±0.013	0.228±0.006	0.648±0.020	0.183±0.020	0.176±0.004	0.562±0.032	6.0***
GMI	[illegible]	[illegible]	0.606±0.011	0.210±0.011	0.238±0.004	0.547±0.008	0.185±0.009	0.199±0.006	0.486±0.015	7.1***
MinCutPool	[illegible]	0.157±0.024	0.677±0.012	0.350±0.130	0.067±0.050	0.385±0.287	0.435±0.136	0.021±0.037	0.154±0.256	8.6***
DMoN	0.346±0.024	0.182±0.017	0.665±0.011	0.308±0.012	0.139±0.009	0.624±0.008	0.283±0.009	0.112±0.006	0.591±0.009	6.6***
GCC	0.410±0.011	0.191±0.031	0.545±0.066	0.415±0.024	0.147±0.039	0.448±0.114	0.419±0.054	0.078±0.039	0.312±0.161	6.3***
GCN-Jaccard	0.369±0.031	0.209±0.018	0.676±0.009	0.337±0.011	0.171±0.008	0.643±0.004	0.306±0.012	0.139±0.007	0.612±0.007	4.0***
GCN-SVD	0.280±0.033	0.120±0.013	0.448±0.021	0.248±0.027	0.084±0.008	0.422±0.022	0.237±0.027	0.062±0.010	0.398±0.022	9.8***
GDC	0.257±0.026	0.117±0.015	0.548±0.021	0.231±0.015	0.096±0.012	0.530±0.013	0.232±0.018	0.089±0.012	0.529±0.014	9.4***
ProGNN	0.359±0.025	0.191±0.017	0.636±0.012	0.326±0.016	0.153±0.009	0.587±0.013	0.302±0.012	0.125±0.006	0.544±0.012	5.9***
PTDNet	0.278±0.029	0.048±0.004	0.344±0.007	0.277±0.044	0.036±0.014	0.317±0.019	0.293±0.037	0.056±0.025	0.301±0.018	11.3***
FGC	0.410±0.004	0.131±0.005	0.409±0.007	0.398±0.005	0.112±0.007	0.381±0.008	0.400±0.005	0.105±0.005	0.370±0.008	7.4***
MetaGC	0.363±0.017	0.230±0.013	0.707±0.007	0.330±0.025	0.194±0.021	0.677±0.012	0.289±0.017	0.151±0.013	0.640±0.009	3.3

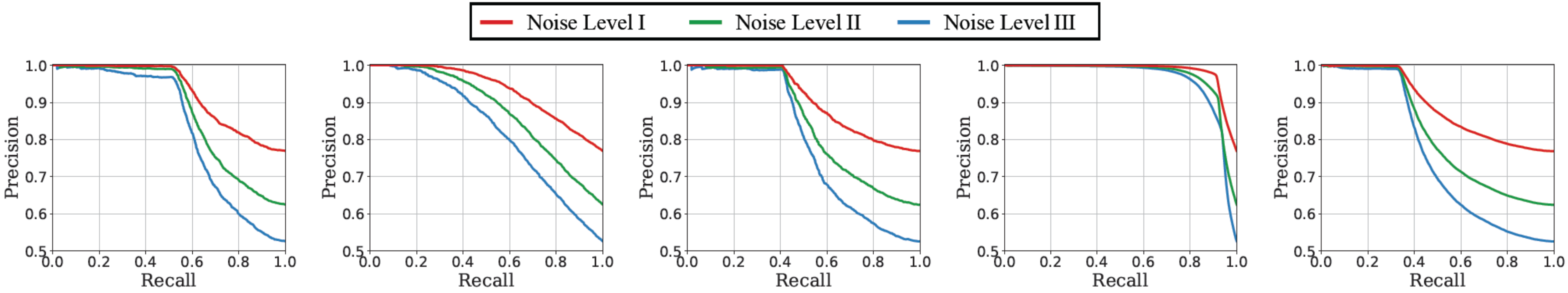
(e) Citeseer

MetaGC achieves
an average rank of 1.7 to 3.3
in the other datasets

(Q2) Effectiveness of Meta-Weighting

- The meta-model **accurately distinguish noise edges** from the real ones.

Dataset	Cora			Cora-ML			Citeseer			Amazon-Photo			Pubmed		
Noise Level	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
Meta-Weighting	0.927	0.875	0.831	0.934	0.878	0.825	0.908	0.843	0.793	0.993	0.985	0.976	0.890	0.813	0.757
Baseline	0.769	0.625	0.526	0.769	0.625	0.526	0.769	0.625	0.526	0.769	0.625	0.526	0.769	0.625	0.526



(a) Cora (b) Cora-ML (c) Citeseer (d) Amazon-Photo (e) Pubmed

(Q3) Ablation Study

- **Every component** of MetaGC **contributes** to the performance gain.

Noise Level	I			II			III		
Metric	F1 Score	NMI	Modularity	F1 Score	NMI	Modularity	F1 Score	NMI	Moduarity
MetaGC-X	0.340±0.022	0.203±0.016	0.695±0.006	0.308±0.017	0.173±0.014	0.662±0.007	0.280±0.018	0.142±0.013	0.634±0.009
MetaGC-A	0.346±0.020	0.214±0.014	0.701±0.007	0.324±0.019	0.187±0.016	0.674±0.011	0.288±0.017	0.150±0.012	0.638±0.009
MetaGC	**0.363±0.017**	**0.230±0.013**	**0.707±0.007**	**0.330±0.025**	**0.194±0.021**	**0.677±0.012**	**0.289±0.017**	**0.151±0.013**	**0.640±0.009**

- **MetaGC-X:** MetaGC **without the meta-model**, i.e., the weights for all node pair are the same.
- **MetaGC-A**: MetaGC with the meta-model **using only node attributes**.

Robust Graph Clustering via Meta Weighting for Noisy Graphs

Introduction

Preliminaries

Method

Experiment

Conclusions

Conclusion

- We propose **MetaGC** for **robust GNN-based graph clustering** against noise edges.
- MetaGC is robust against noise edges, achieving the **best clustering performance overall** among all the 14 considered methods.
- The meta-model in MetaGC assigns **high weights to real edges** and **low weights to noise edges**, leading to MetaGC's performance gain.

Source Code & Dataset: https://github.com/HyeonsooJo/MetaGC