A Provable Framework of Learning Node Embeddings via Graph Summarization

Houquan Zhou[1], Shenghua Liu[1], Danai Koutra[2], Huawei Shen[1], Xueqi Cheng[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences

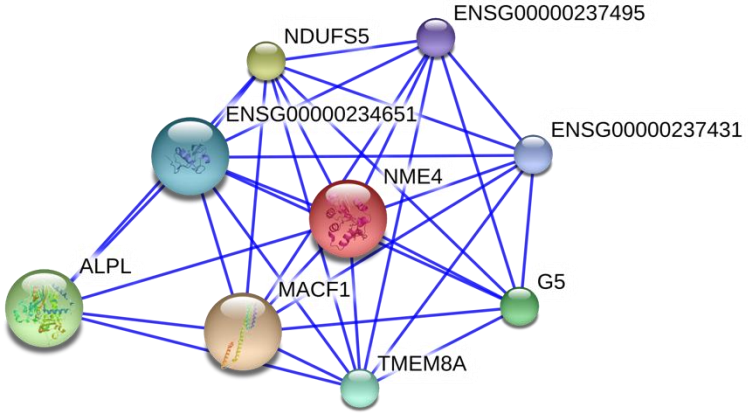
[2] University of Michigan

Outline

- **Introduction**
- Theory & Framework
- Experiments
- Conclusion

Graphs are everywhere

Social Network



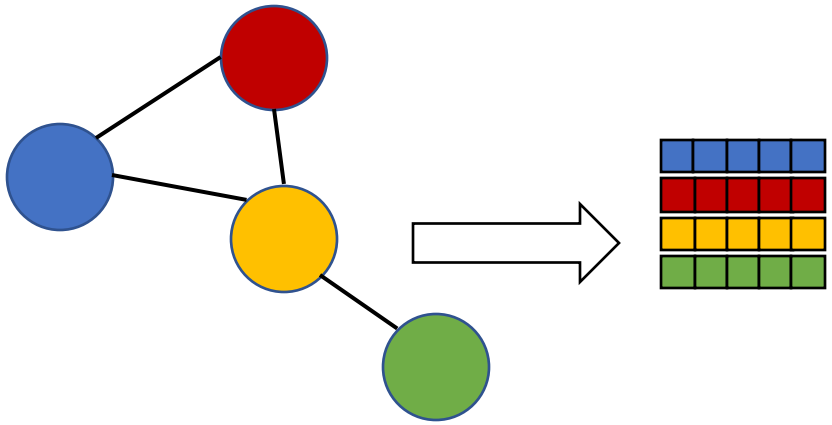
Protein Interaction Network

Money Transaction Network

Graphs are widely used to represent relationship between objects in various domains.

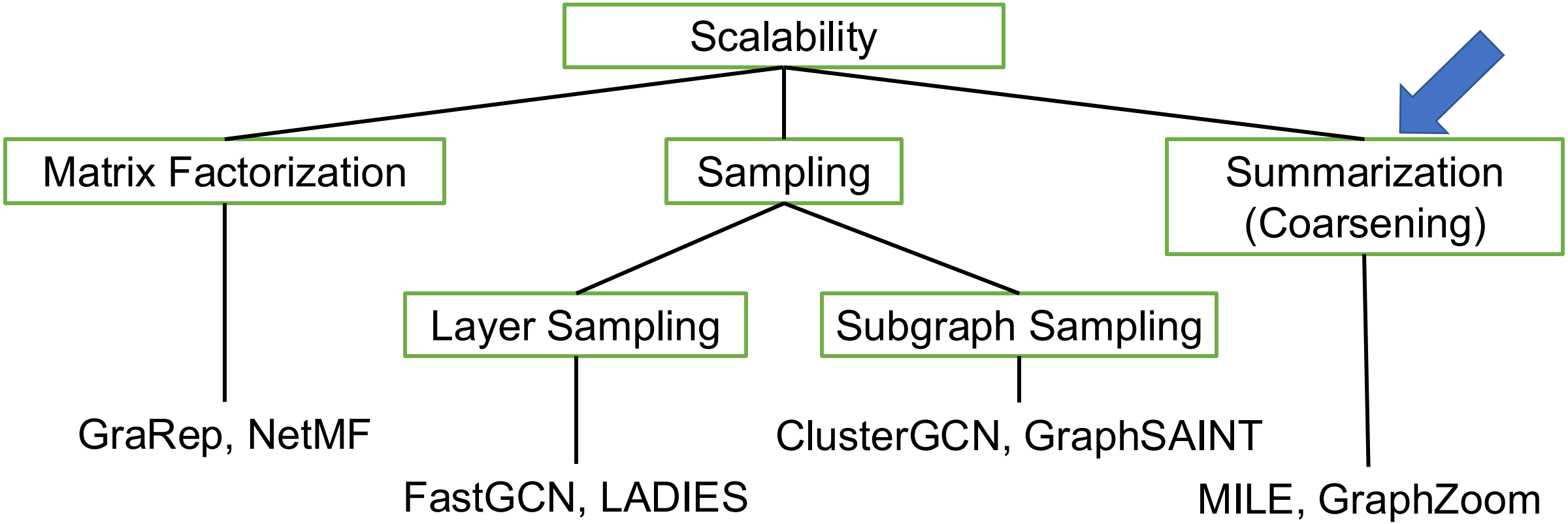
Node Embedding

- Node embedding is a fundamental problem in graph mining.
- Node embedding methods map each node in graphs to a low-dimensional vector, called embeddings.
- Embeddings can capture structural information of nodes.
- Embeddings can be used in subsequent tasks, e.g., node classification, link prediction, anomaly detection, etc.



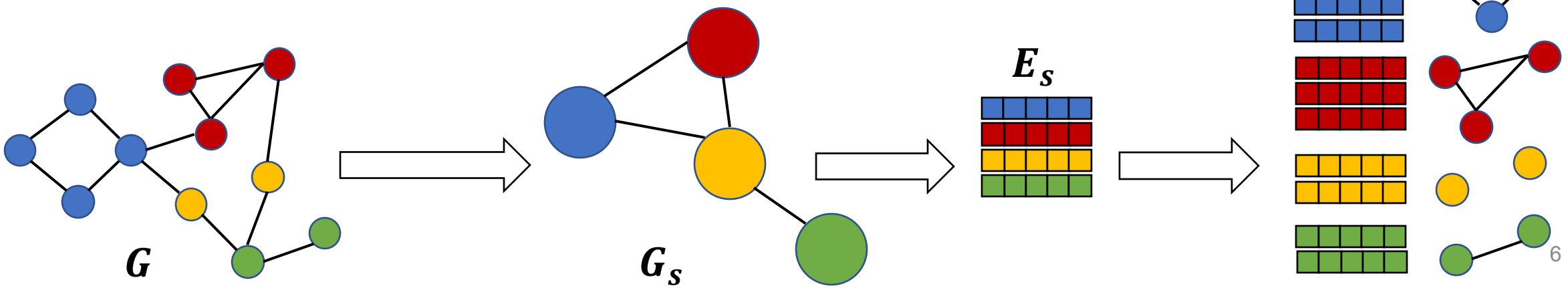
Scalability

- As the network size goes larger, the scalability of embedding learning methods is a key challenge.



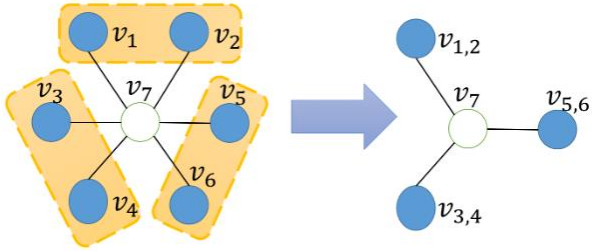
Framework of learning embeddings via graph summarization

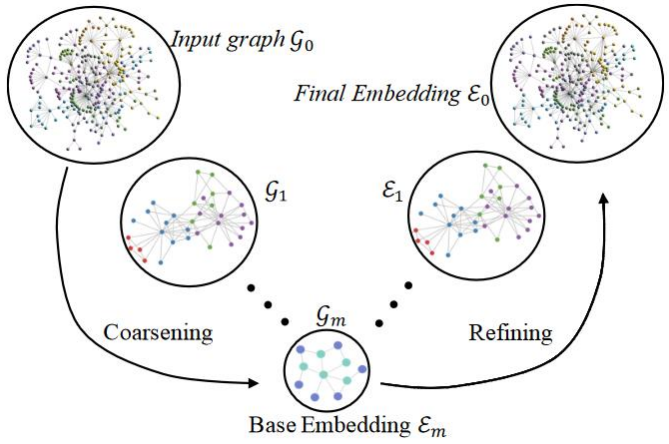
- S1: Summarize original graphs G to smaller summary graphs G_s (typically by merging nodes into supernodes)
- S2: Learn embeddings on summary graphs (faster due to the smaller size)
- S3: Restore to original embeddings (typically copy)
- S4: (Optional) Refine the embeddings

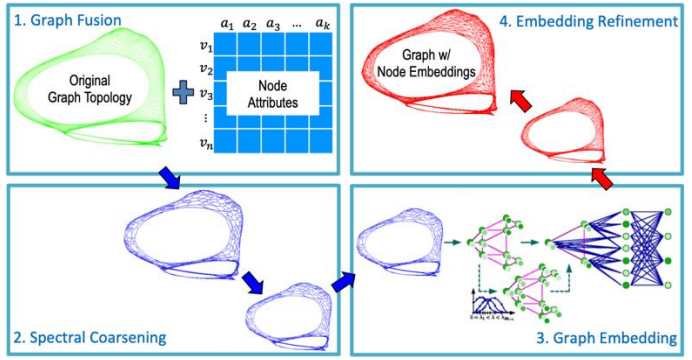


Summarization for learning embeddings

- HARP '18
- MILE '18
- GraphZoom '20



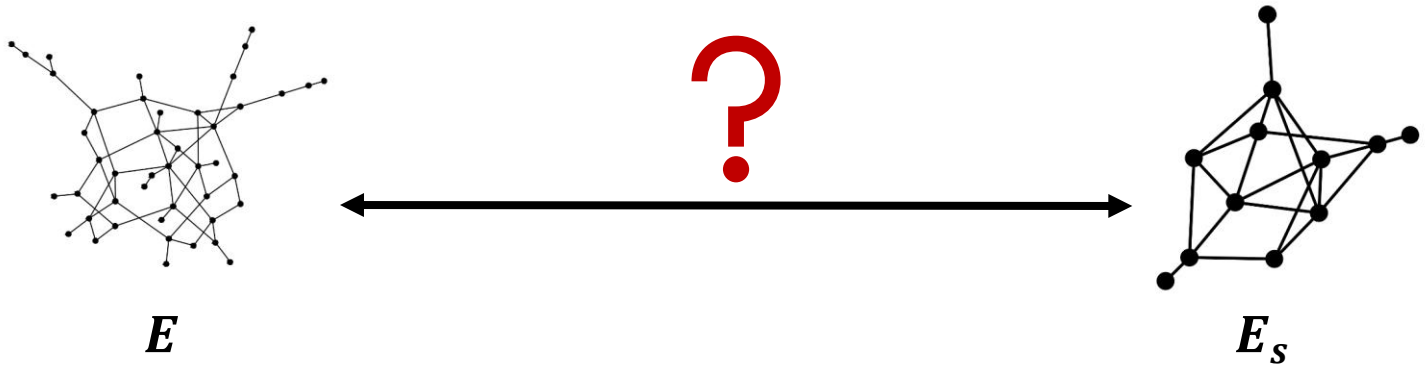




- They summarize input graphs and restore embeddings **empirically**, and do not investigate **the underlying mechanism**.

Question

- What is the connection between embeddings learned on original graphs and summary graphs?
- Can we find out the theoretical connection between them?

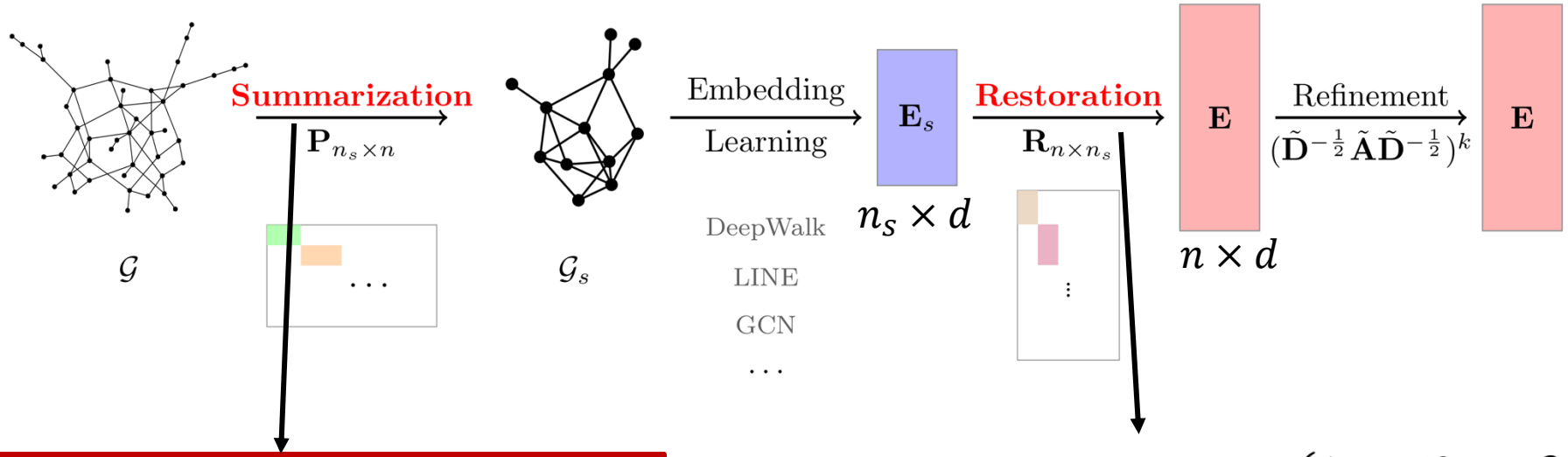


Main Contributions

- We give theoretical analysis about learning node embeddings via graph summarization.
- Three node embedding methods: DeepWalk, LINE and GCN.
- Answer two questions:
  - How should we summarize the input graph?
  - How should we restore original embeddings from summary embeddings?

Main Contributions

- We propose a framework, GELSumm, to learn node embeddings via graph summarization.



Goal: Make G and G_r close.

DeepWalk & LINE: $\mathbf{R}(i,p) = \begin{cases} 1 & if\ v_i \in \mathcal{S}_p \\ 0 & otherwise. \end{cases}$

GCN: $\mathbf{R}(i,p) = \begin{cases} \sqrt{\frac{d_i}{d_p^{(s)}}} & if\ v_i \in \mathcal{S}_p \\ 0 & otherwise. \end{cases}$

Main Contributions

- GELSumm can learn high-quality embeddings with less time.

Up to 8.78x faster with 2.34% performance gain.

Table 3: Node classification results(DeepWalk & LINE). Average running times and accuracy scores over 10 runs are reported. "orig" represents the results on original graphs. r stands for the relative node size. The running time includes the summarization time, the embedding learning time and the refinement time. It can be observed that GELSUMM-DeepWalk and GELSUMM-LINE obtain better results than original DeepWalk and LINE using much less time.

		DeepWalk				LINE			
		orig	r=0.6	r=0.4	r=0.2	orig	r=0.6	r=0.4	r=0.2
Cora	acc (%)	72.11	73.29	73.87	**74.67**	68.27	66.37	65.63	**68.79**
	time (secs)	107.274	59.452	39.334	**17.255**	14.246	9.313	6.233	**3.172**
Citeseer	acc (%)	46.24	48.87	48.61	**49.08**	41.36	44.81	45.15	**46.13**
	time (secs)	111.022	67.802	42.414	**12.641**	16.991	7.885	5.305	**3.376**
Pubmed	acc (%)	72.35	73.31	73.93	**74.05**	68.51	69.74	**71.40**	69.86
	time (secs)	875.838	523.848	359.602	**175.897**	112.067	65.563	50.331	**31.578**
Flickr	acc (%)	53.19	**53.36**	53.02	52.87	51.47	**52.42**	52.26	50.72
	time (secs)	6142.00	3203.27	2095.68	**1439.87**	528.04	270.05	168.74	**92.12**

Outline

- Introduction
- **Theory & Framework**
- Experiments
- Conclusion

Kernel matrix

- Analyzed methods: DeepWalk, LINE, and GCN
- To analyze them simultaneously, we first unify them with the following kernel matrix form:

$$\mathcal{K}(\mathcal{G}) := (D^{-c}AD^{-1+c})^{\tau} D^{1-2c}$$

$c = 1$: $(D^{-1}A)^{\tau}\, D^{-1}$

$c = \frac{1}{2}$: $(D^{-\frac{1}{2}}\, A\, D^{-\frac{1}{2}})^{\tau}$

Method $\mathcal{M}$	Kernel matrix $\mathcal{K}(\mathcal{G})$
DeepWalk	$(\mathbf{D}^{-1}\mathbf{A})^{\tau}\mathbf{D}^{-1}$
LINE	$\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}$
GCN	$\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$

[Qiu2018]

Main conclusion: Kernel matrix approximation

- Learning embeddings on original graphs: $\mathcal{K}(\mathcal{G})$
- Learning embeddings on summary graphs: $\mathcal{K}(\mathcal{G}_s)$
- We prove the following relationship between them:

$$\mathcal{K}(\mathcal{G}) := \left(\mathbf{D}^{-c}\mathbf{A}\mathbf{D}^{-1+c}\right)^{\tau}\mathbf{D}^{1-2c}$$

$$\approx \left(\mathbf{D}^{-c}\mathbf{A}_r\mathbf{D}^{-1+c}\right)^{\tau}\mathbf{D}^{1-2c} = \mathcal{K}(\mathcal{G}_r) = \mathbf{R}\,\mathcal{K}(\mathcal{G}_s)\,\mathbf{R}^{\mathrm{T}}$$

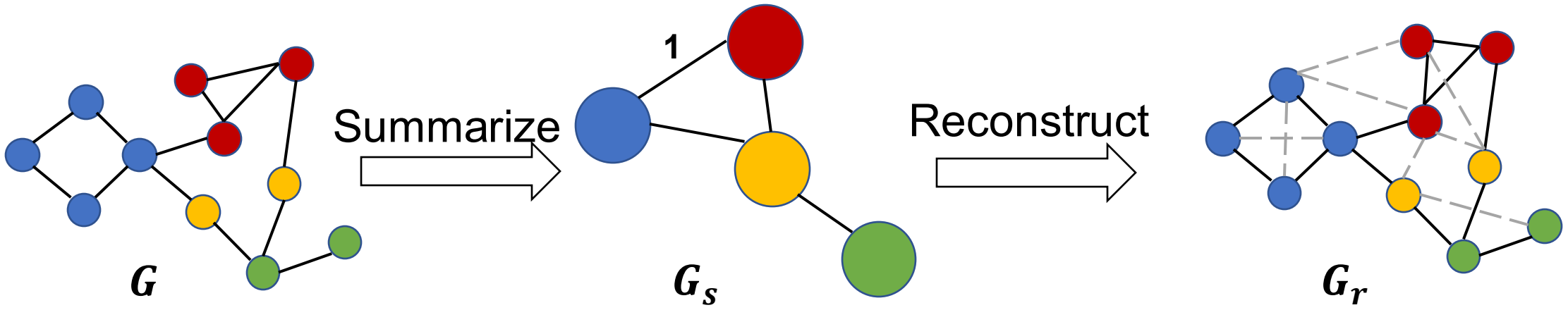
Reconstructed graph

$$\mathbf{R}(i,p) = \begin{cases} \left(\frac{d_i}{d_p^{(s)}}\right)^{1-c} & \text{if } v_i \in \mathcal{S}_p \\ 0 & \text{otherwise,} \end{cases}$$

- Detailed proof can be found in the paper.

Proof: Reconstructed graph

- Detailed connection between supernodes are lost.
- Reconstruct edges according to a predefined scheme.
- Use the configuration-based reconstruction in our analysis.



Proof: Reconstructed graph

- Configuration-based reconstruction is based on the configuration random graph model used in modularity [Newman 2010]

$$\mathbf{A}_r(i,j) = \frac{d_i}{d_p^{(s)}} \mathbf{A}_s(p,q) \frac{d_j}{d_q^{(s)}} \qquad v_i \in \mathcal{S}_p, v_j \in \mathcal{S}_q$$

Node j's degree proportion in supernode S_q

Connection weight between supernode S_p and S_q

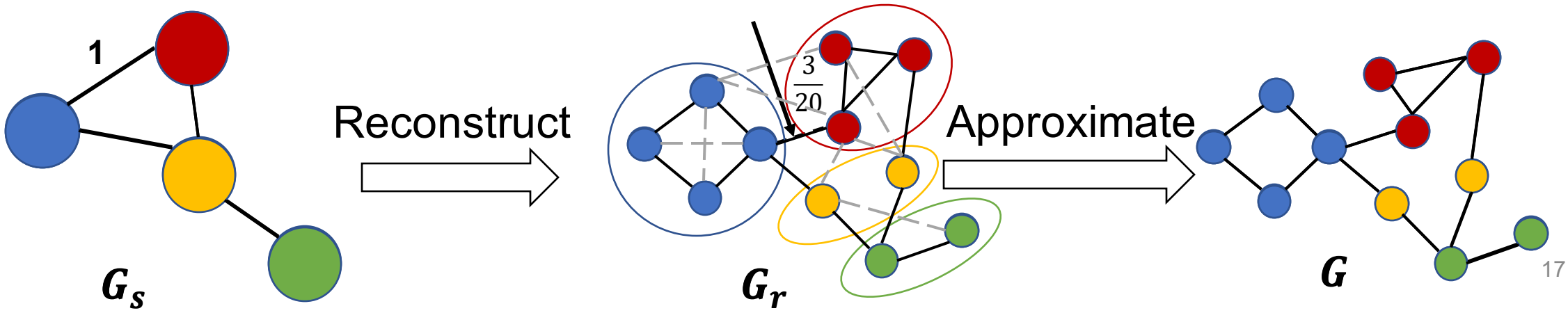
Node i's degree proportion in supernode S_p

node i in supernode S_p, node j in supernode S_q

Proof: Reconstructed graph

$$\mathbf{A}_r(i,j) = \frac{d_i}{d_p^{(s)}} \mathbf{A}_s(p,q) \frac{d_j}{d_q^{(s)}} \quad v_i \in \mathcal{S}_p, v_j \in \mathcal{S}_q$$

$$\frac{3}{20} = \frac{4}{10} \times 1 \times \frac{3}{8}$$

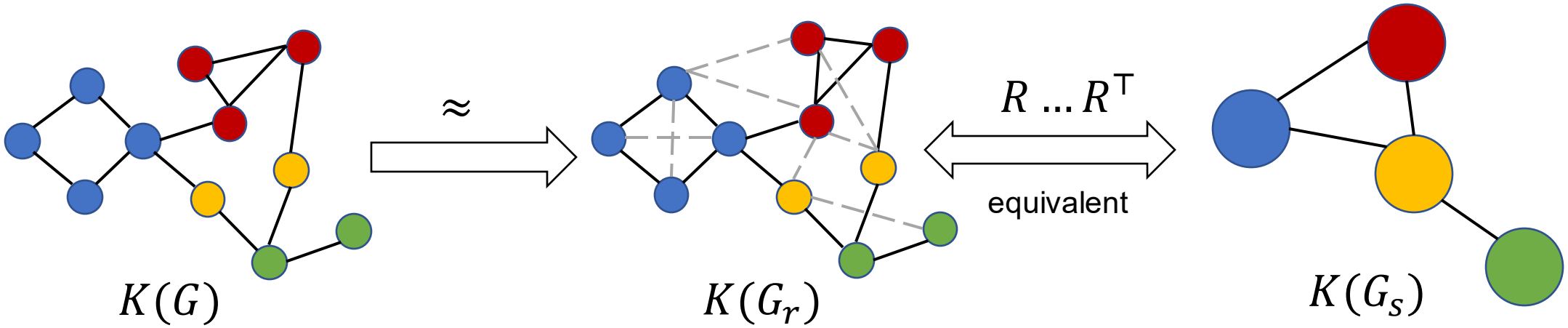


Proof: Kernel matrix approximation

$$\mathcal{K}(\mathcal{G}) := \left(\mathbf{D}^{-c}\mathbf{A}\mathbf{D}^{-1+c}\right)^{\tau}\mathbf{D}^{1-2c}$$

$$\approx \left(\mathbf{D}^{-c}\mathbf{A}_r\mathbf{D}^{-1+c}\right)^{\tau}\mathbf{D}^{1-2c} = \mathcal{K}(\mathcal{G}_r) = \mathbf{R}\ \mathcal{K}(\mathcal{G}_s)\ \mathbf{R}^{\mathrm{T}}$$

Make A and A_r close.



Learning on the original graph.

Learning on the reconstructed graph.

Learning on the summary graph.

Detailed proof can be found in the paper.

GELSumm for DeepWalk & LINE

- $c = 1$: DeepWalk and LINE.
- Original embeddings E can be approximated by RE_S.

$$\mathcal{K}(\mathcal{G}) = \left(\mathbf{D}^{-1}\mathbf{A}\right)^{\tau}\mathbf{D}^{-1} \approx \left(\mathbf{D}^{-1}\mathbf{A}_r\right)^{\tau}\mathbf{D}^{-1}$$
$$= \mathbf{R}\left(\mathbf{D}_s^{-1}\mathbf{A}_s\right)^{\tau}\mathbf{D}_s^{-1}\mathbf{R}^{\mathrm{T}},$$

where

$$\mathbf{R}(i,p) = \begin{cases} 1 & \text{if } v_i \in \mathcal{S}_p \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 1. *Embeddings learned by DeepWalk on the original graph $\mathcal{G}$, $\mathbf{E}$, can be approximated by embeddings learned by DeepWalk on the summary graph $\mathcal{G}_s$, $\mathbf{E}_s$, using the restoration matrix $\mathbf{R}$ in (10), i.e.,*

$$\mathbf{E} \approx \mathbf{R}\,\mathbf{E}_s \tag{13}$$

Interpretation: Supernode embeddings copied to node embeddings.

Detailed proof can be found in the paper.

GELSumm for GCN

- $c = \frac{1}{2}$: GCN.
- Original embeddings E can be approximated by RE_s.

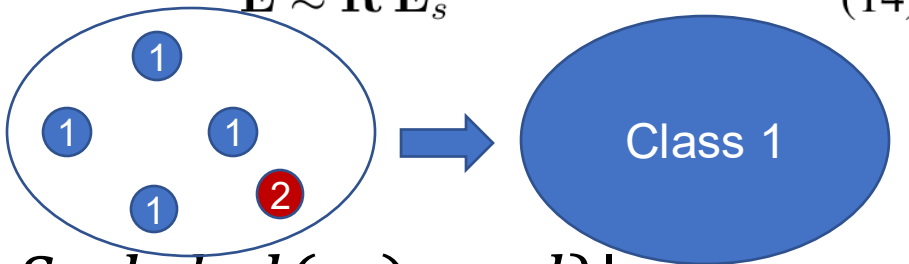
$$\mathcal{K}(\mathcal{G}) = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \approx \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}}$$
$$= \mathbf{R}\left(\mathbf{D}_s^{-\frac{1}{2}}\mathbf{A}_s\mathbf{D}_s^{-\frac{1}{2}}\right)\mathbf{R}^{\mathrm{T}},$$

where

$$\mathbf{R}(i,p) = \begin{cases} \sqrt{\frac{d_i}{d_p^{(s)}}} & \text{if } v_i \in \mathcal{S}_p \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 2. *Embeddings learned by GCN on the original graph $\mathcal{G}$ can be approximated by embeddings learned by GCN on the summary graph $\mathcal{G}_s$ with initial features $\mathbf{X}_s := \mathbf{R}^{\mathrm{T}}\mathbf{X}$, using the restoration matrix $\mathbf{R}$ defined in (12), in a **least-square approximation** perspective:*

$$\mathbf{E} \approx \mathbf{R}\,\mathbf{E}_s \qquad (14)$$

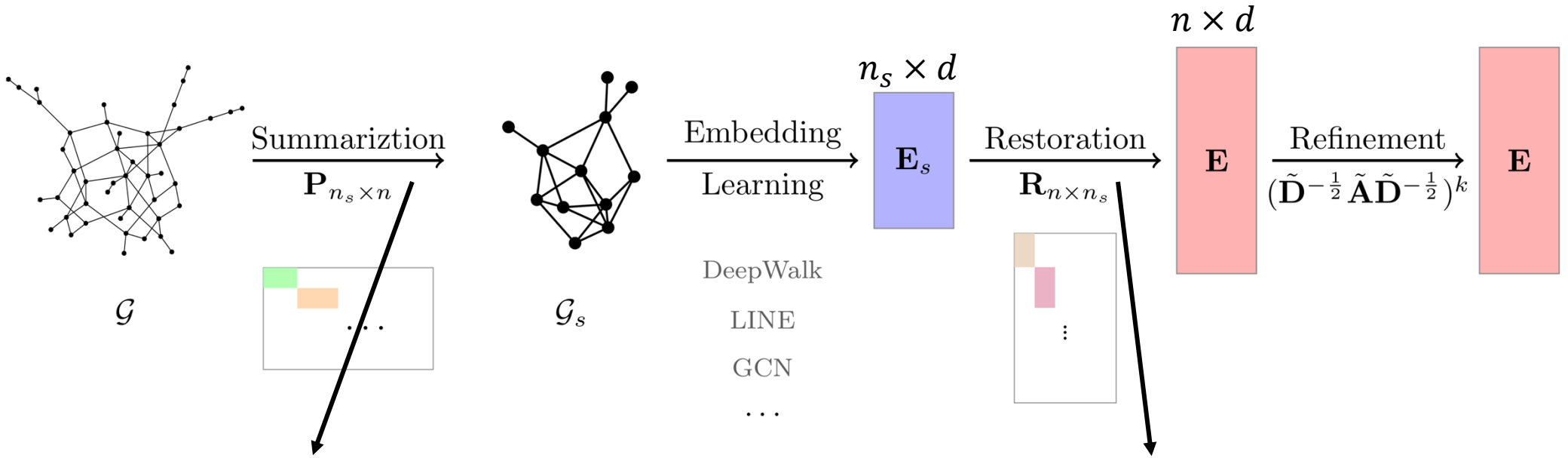


- Label of supernodes: majority vote.

$$label(S_p) = argmax_l\, |\{v_i | v_i \in S_p, label(v_i) = l\}|$$

Detailed proof can be found in the paper.

GELSumm Framework



Q: How to summarize?

A: Make G and G_r close.

Reconstructed from G_s according to the configuration model.

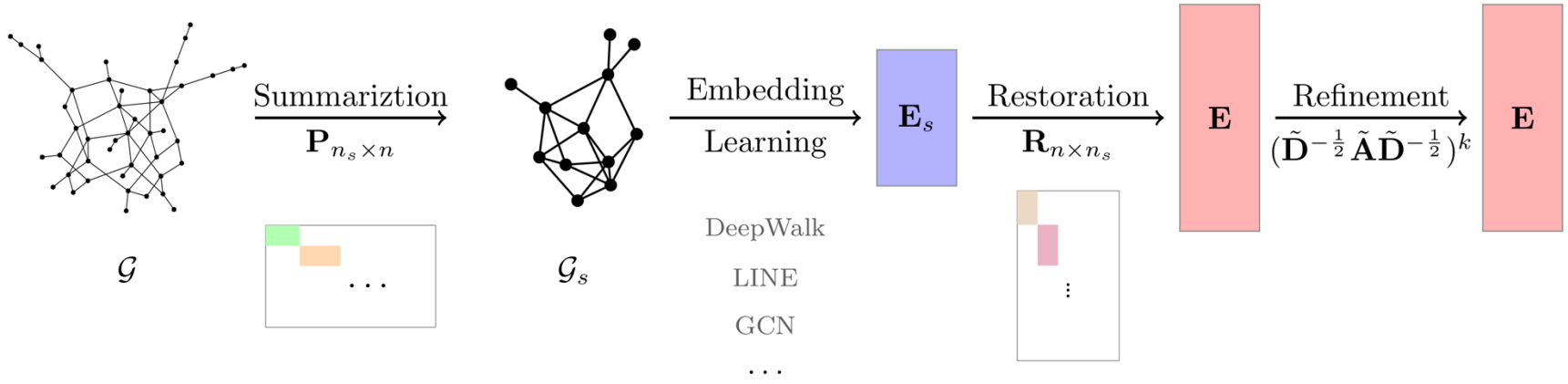
Q: How to restore?

DeepWalk & LINE: $\mathbf{R}(i,p) = \begin{cases} 1 & if\ v_i \in \mathcal{S}_p \\ 0 & otherwise. \end{cases}$

GCN: $\mathbf{R}(i,p) = \begin{cases} \sqrt{\frac{d_i}{d_p^{(s)}}} & if\ v_i \in \mathcal{S}_p \\ 0 & otherwise. \end{cases}$

Four steps

- S1: Summarization. Using DPGS[Zhou2021], a configuration-based summarization method.
- S2: Learn embeddings.
- S3: Restore the embeddings using R matrix.
- S4: Refinement using a low-pass filter $\left(\tilde{D}^{-\frac{1}{2}}A\tilde{D}^{-\frac{1}{2}}\right)^{k}$ [Deng2020].



Outline

- Introduction
- Theory & Framework
- **Experiments**
- Conclusion

Settings

- Datasets
  - Citation networks: cora, citeseer, pubmed
  - Social networks: flickr, reddit
  - Co-purchase network: amazon2M
- Baselines:
  - HARP
  - MILE
  - GraphZoom
- Summarization ratios: [0.6, 0.4, 0.2]

GELSumm for DeepWalk & LINE

- GELSumm can learn high-quality embeddings with less time (up to 8.78x).

Table 3: Node classification results(DeepWalk & LINE). Average running times and accuracy scores over 10 runs are reported. "orig" represents the results on original graphs. r stands for the relative node size. The running time includes the summarization time, the embedding learning time and the refinement time. It can be observed that GELSUMM-DeepWalk and GELSUMM-LINE obtain better results than original DeepWalk and LINE using much less time.

		DeepWalk				LINE			
		orig	r=0.6	r=0.4	r=0.2	orig	r=0.6	r=0.4	r=0.2
Cora	acc (%)	72.11	73.29	73.87	**74.67**	68.27	66.37	65.63	**68.79**
	time (secs)	107.274	59.452	39.334	**17.255**	14.246	9.313	6.233	**3.172**
Citeseer	acc (%)	46.24	48.87	48.61	**49.08**	41.36	44.81	45.15	**46.13**
	time (secs)	111.022	67.802	42.414	**12.641**	16.991	7.885	5.305	**3.376**
Pubmed	acc (%)	72.35	73.31	73.93	**74.05**	68.51	69.74	**71.40**	69.86
	time (secs)	875.838	523.848	359.602	**175.897**	112.067	65.563	50.331	**31.578**
Flickr	acc (%)	53.19	**53.36**	53.02	52.87	51.47	**52.42**	52.26	50.72
	time (secs)	6142.00	3203.27	2095.68	**1439.87**	528.04	270.05	168.74	**92.12**

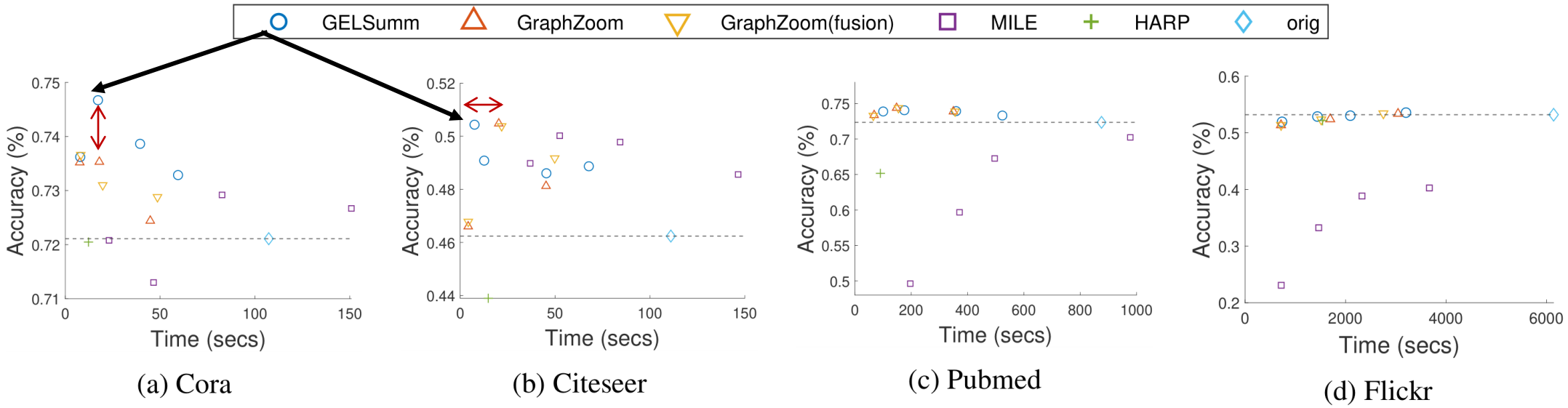
GELSumm for GCN

- For GCN, the performance drops slightly.

Table 5: Node classification results (GCN) on five datasets. "orig" represents the results on original graphs. Using proposed GELSUMM restoration method yields consistently better results than empirically restoration method.

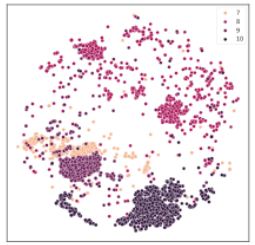
		Cora	Citeseer	Pubmed	Flickr	Reddit
orig	acc(%)	80.20	69.70	78.02	52.89	94.55
	time(secs)	1.92	2.49	5.22	30.77	476.31
$r = 0.6$	GELSumm acc(%)	80.90	70.46	77.38	50.15	93.62
	Empirical acc(%)	80.64	69.94	76.88	48.75	92.55
	time(secs)	1.232	1.4	1.91	22.42	345.72
$r = 0.4$	GELSumm acc(%)	77.76	67.62	76.98	49.80	93.11
	Empirical acc(%)	77.40	66.18	76.6	48.76	87.91
	time(secs)	1.14	1.28	1.65	87.91	292.44
$r = 0.2$	GELSumm acc(%)	75.22	67.64	73.92	49.51	91.90
	Empirical acc(%)	75.24	67.64	73.72	46.15	87.97
	time(secs)	**1.01**	**1.12**	**1.33**	**13.16**	**253.26**

Comparison to baselines

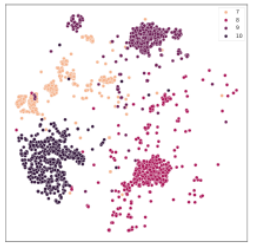


- Better or comparable performance with similar or less time.

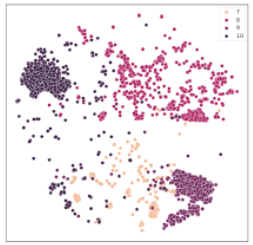
Large dataset: amazon2M



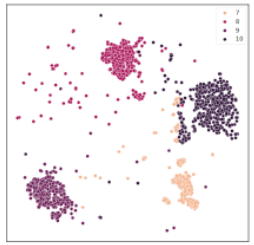
(a) Deepwalk-0.2 accuracy: 0.7441



(b) DeepWalk-0.1 accuracy: 0.7757

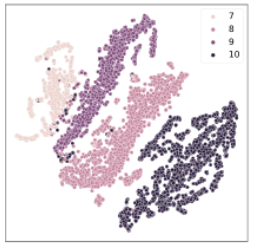


(c) LINE-0.2 accuracy: 0.7597



(d) LINE-0.1 accuracy: 0.7586

(e) GCN-0.2 accuracy: 0.8389



(f) GCN-0.1 accuracy: 0.8321

- Plot some node embeddings using t-sne.
- Nodes from different class are well-separated ⇒ high-quality embeddings.

Outline

- Introduction
- Theory & Framework
- Experiments
- **Conclusion**

Summary

- Study the problem of learning node embeddings via graph summarization and propose a framework GELSumm.
- Properties
  - Theoretical-grounded
  - Flexible
  - Effective
- Future work
  - More graph mining problems
  - More general kernel form

Thank You!

- Email: zhouhouquan18z@ict.ac.cn

- Code: https://github.com/BGT-M/GELSumm

References

- Qiu, Jiezhong, et al. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec." *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018.
- Zhou, Houquan, et al. "DPGS: degree-preserving graph summarization." Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, 2021.
- Newman, M. Networks: An Introduction. OUP Oxford, 2010.
- Deng, Chenhui, et al. "GraphZoom: A Multi-level Spectral Approach for Accurate and Scalable Graph Embedding." International Conference on Learning Representations. 2019.