Zero-shot Node Classification with Decomposed Graph Prototype Network

Zheng Wang, Jialong Wang, Yuchen Guo, Zhiguo Gong

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

北京科技大学
University of Science and Technology Beijing

SKL-IOTSC
智慧城市物聯網國家重點實驗室(澳門大學)
Laboratório de Referência do Estado de Internet das Coisas para a Cidade Inteligente (Universidade de Macau)
State Key Laboratory of Internet of Things for Smart City (University of Macau)

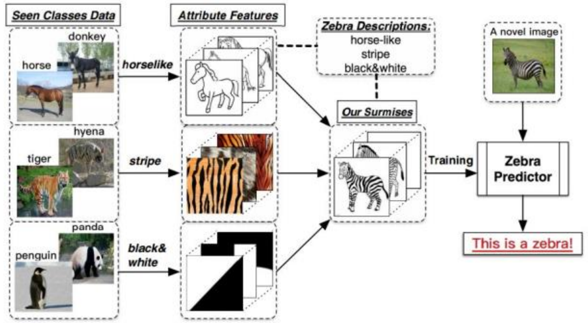
KDD 2021

https://github.com/zhengwang100/dgpn

Outline

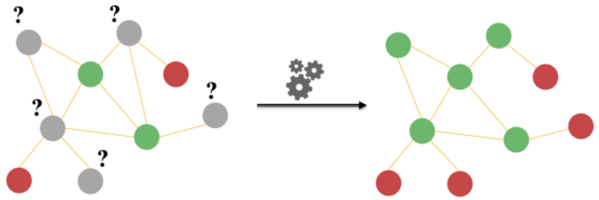
- **Motivation and Problem**
  - Related work (Zero-shot & Node classification)
  - The problem of Zero-shot Node Classification (ZNC)
- Our solution
  - Step I: Acquiring High-Quality CSDs
  - Step II: Designing well-generalized graph-based learning models
- Experiments

Related Works



- Zero-shot learning

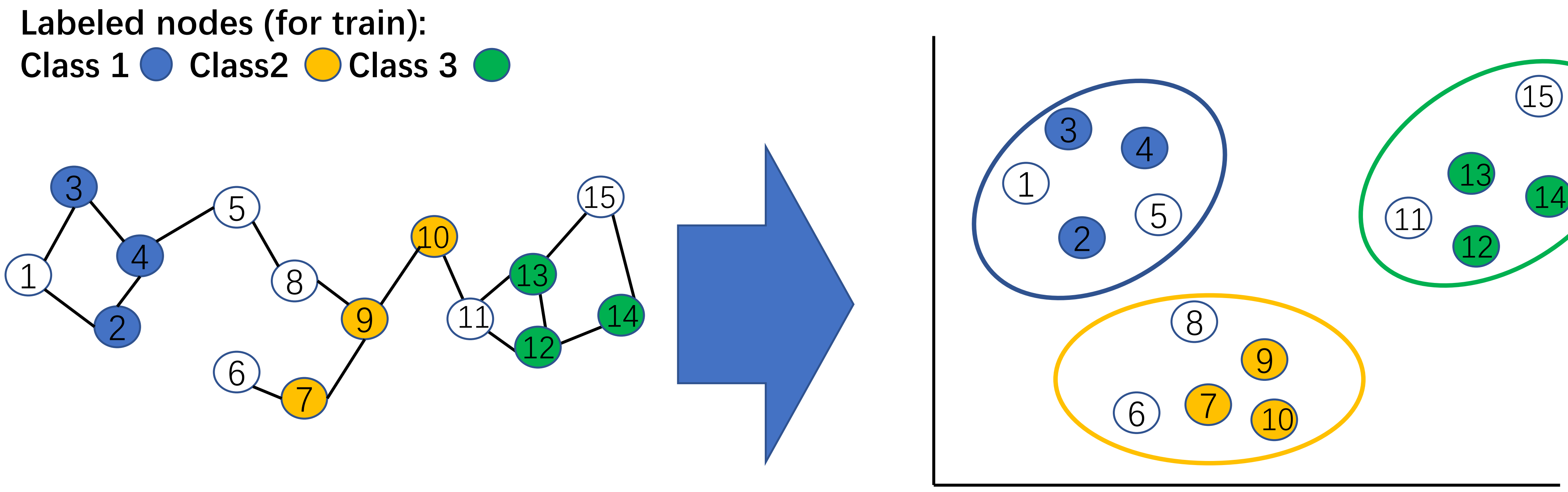
  Classify the samples belonging to the classes that have no labeled data. Most ZSL methods are based on the external description and human-made attributes. Limited to computer vision or natural language processing.



- Graph Node Classification

  The method of processing the graph is divided into the early shallow method and the recent deep Graph neural network method. Nevertheless, existing methods generally all assume that every class in the graph has some labeled nodes.

Traditional Node Classification

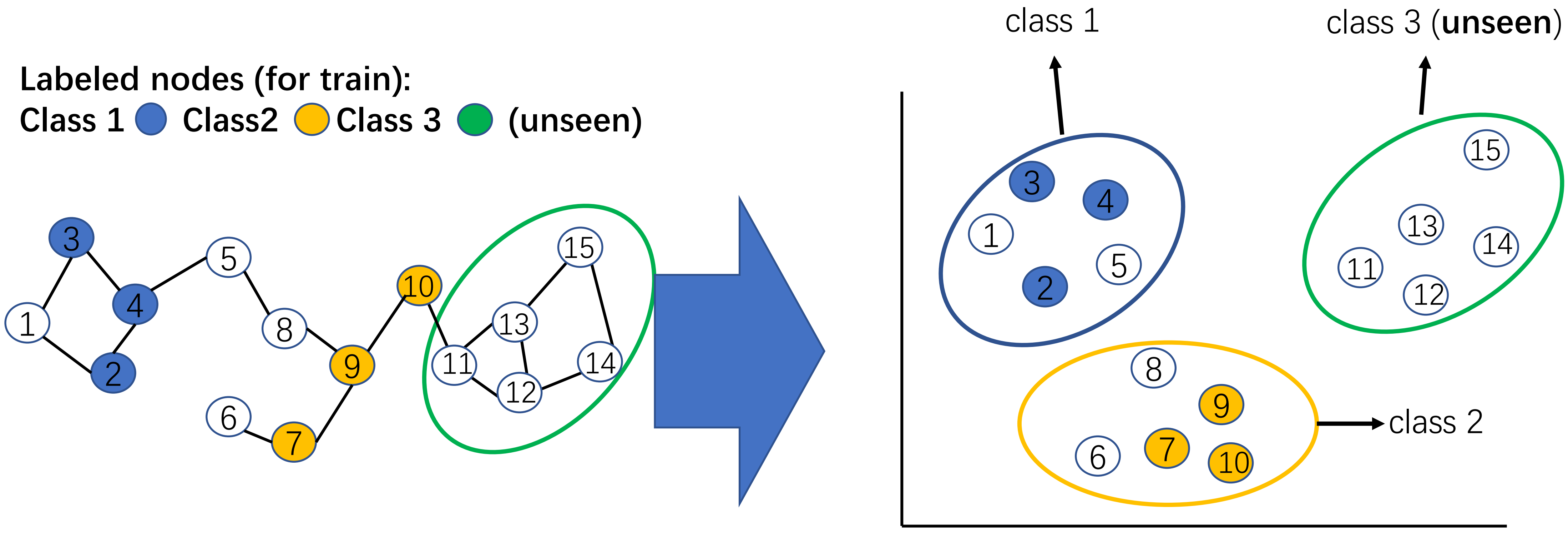


(a) Input: graph and labels

(b) Output: predict labels

We have input with graph and corresponding labeled nodes for **every class**, and our goal is to predict labels on the unlabeled nodes.

Zero-shot Node Classification (ZNC)



(a) Input: graph and labels

(b) Output: predict labels

Although class 3 has no labeled samples for training (i.e., the zero-shot setting), we still want to "find" those nodes belonging to this class.

Outline

- Motivation and Problem
  - Background or motivation
  - The problem of Zero-shot Node Classification (ZNC)
- Our solution
  - **Step I: Acquiring High-Quality CSDs**
  - Step II: Designing well-generalized graph-based learning models
- Experiments

Step I: Acquiring High-Quality CSDs

- Acquiring two kinds (candidates) of CSDs from Wikipedia

Class name	Class description (wiki page)
DB	In computing, a database is an organized collection of data stored and accessed electronically from a computer system. Where…….
ML	Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data….
AI	Artificial intelligence (AI) is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and……
… …	… …

Word2Vec model: Bert

Label-CSDs | **Text-CSDs**

Step I: Acquiring High-Quality CSDs

- Evaluate the quality of the automatically generated CSDs

Empirical probability
(generated from class center representations)

$$Pr(c_j|c_i) = \frac{exp(o_i^\top \cdot o_j)}{\sum_{t,t\neq i}^{|C|} exp(o_i^\top \cdot o_t)}$$

Probability
(generated from our CSDs' vectors)

$$\hat{P}r(c_j|c_i) = \frac{exp(s_i^\top \cdot s_j)}{\sum_{t,t\neq i}^{|C|} exp(s_i^\top \cdot s_t)}$$

Calculate the distance: $$\frac{1}{|C|} \sum_{c_i \in C} dis(Pr(\cdot|c_i), \hat{P}r(\cdot|c_i))$$

Step I: Acquiring High-Quality CSDs

- CSDs' Evaluation Results

Table 1: Quality of obtained CSDs.

Dataset	CSDs Type	KL Divergence ↓	Cosine Similarity ↑	Euclidean Distance ↓
Cora	LABEL-CSDs	0.0154	0.9978	0.1787
	TEXT-CSDs	**0.0109**	**0.9985**	**0.1552**
Citeseer	LABEL-CSDs	0.0120	0.9980	0.1620
	TEXT-CSDs	**0.0077**	**0.9987**	**0.1328**
C-M10M	LABEL-CSDs	0.0062	0.9990	0.1175
	TEXT-CSDs	**0.0026**	**0.9996**	**0.0735**

Here, '↓' indicates the lower the better, whereas '↑' indicates the higher the better.

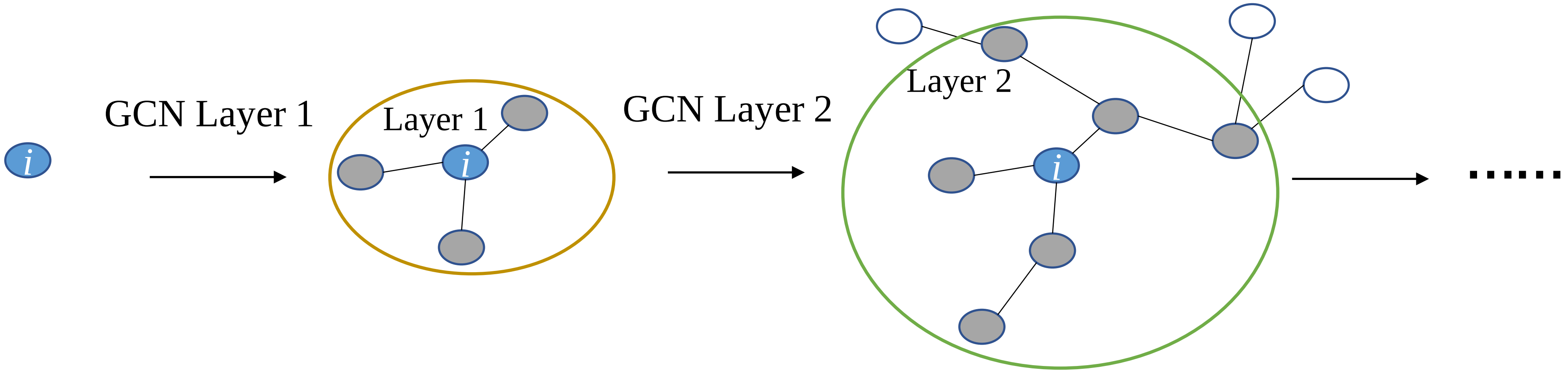
Compared with the Label-CSDs, Text-CSDs always perform better.

Outline

- Motivation and Problem
  - Background or motivation
  - The problem of Zero-shot Node Classification (ZNC)
- Our solution
  - Step I: Acquiring High-Quality CSDs
  - **Step II: Designing well-generalized graph-based learning models**
- Experiments

Step II: Designing well-generalized graph-based learning models

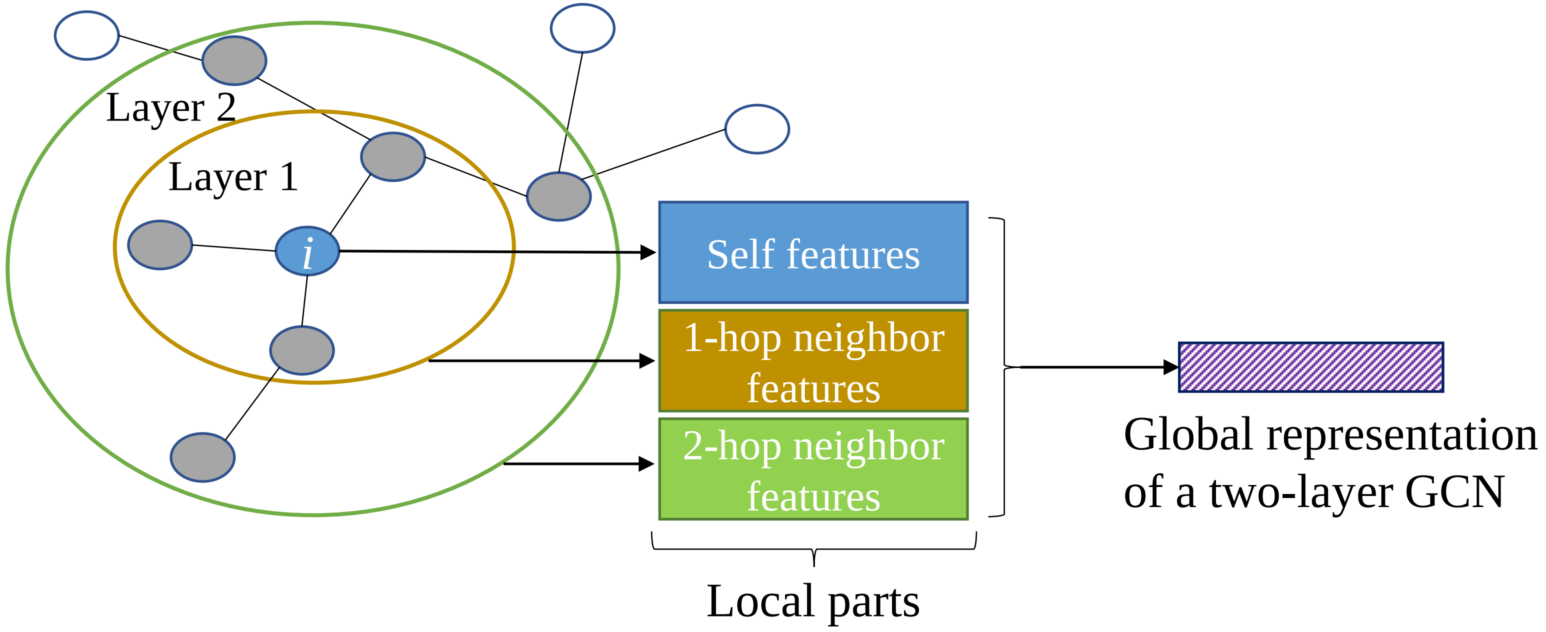
- Traditional GCNs



Traditional GCN processes the input serially.

Step II: Designing well-generalized graph-based learning models

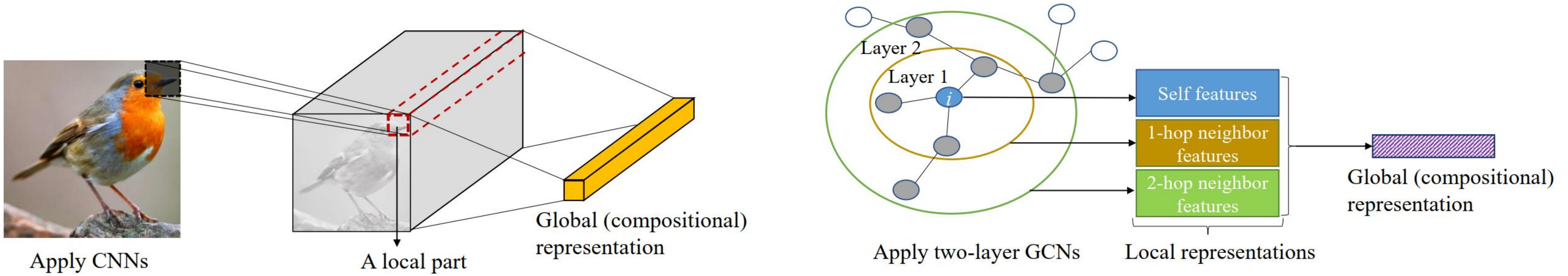
- GCNs Decomposition



We Decompose the two layers GCN into three parts and use them for subsequent locality and compositionality.

Step II: Designing well-generalized graph-based learning models

- Locality and Compositionality



(a) Apply CNNs to an image: for this image, the local feature refers to the representations learned from a "patch" of an image, and the global feature refers to the pooling (like concatenation) result of those local ones.

(b) Apply K-layer (here K=2) GCNs to a graph: for a node in this graph, the $\{k\}_{k=0}^{K}$-th local feature of this node refers to its k-hop neighbor information, and the global feature refers to the weighted sum of all its 3 (i.e., $K + 1$) local ones.

Figure 5: CNNs for images v.s. GCNs for graphs

We take the feature of each order neighbor of the node as the local part and use their combination as the global compositional part.

Decomposed Graph Prototype Network

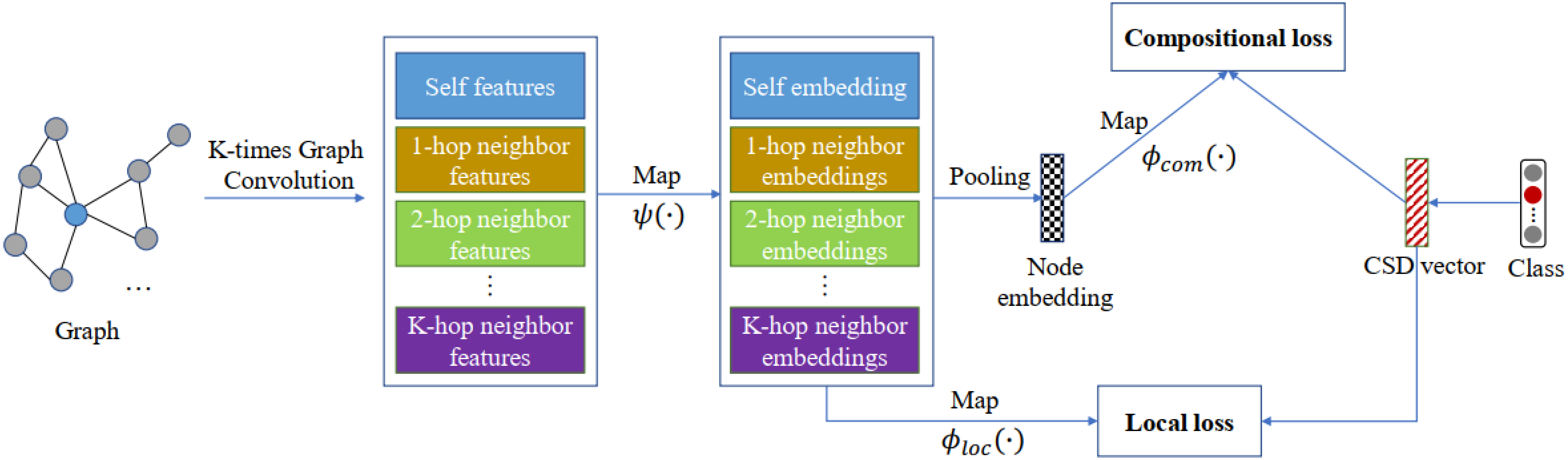


Figure 2: The architecture of Decomposed Graph Prototype Network (DGPN).

This joint learning not only enhances the locality of the node representation that is critical for zero-shot generalization, but also guarantees the discriminability of the global compositional representation for the final node classification.

Outline

- Motivation and Problem
  - Related work
  - The problem of Zero-shot Node Classification (ZNC)
- Our solution
  - Step I: Acquiring High-Quality CSDs
  - Step II: Designing well-generalized graph-based learning models
- **Experiments**

Experimental Settings

- Datasets with seen/unseen splits:

Dataset	Nodes	Edges	Features	Classes	Class Split I [Train/Val/Test]	Class Split II [Train/Val/Test]
Cora	2,708	5,429	1,433	7	[3/0/4]	[2/2/3]
Citeseer	3,327	4,732	3,703	6	[2/0/4]	[2/2/2]
C-M10M	4,464	5,804	128	6	[3/0/3]	[2/2/2]

- Baselines
  - DAP & DAP(CNN)
  - ESZSL
  - ZS-GCN & ZS-GCN(CNN)
  - WDVSc
  - Hyperbolic-ZSL
  - RandomGuess

Compare with Baselines

Table 3: Zero-shot node classification accuracy (%).

		Cora	Citeseer	C-M10M
Class Split I	RandomGuess	25.35±1.28	24.86±1.63	33.21±1.08
	DAP	26.56±0.37	<u>34.19±0.97</u>	<u>39.20±0.54</u>
	DAP(CNN)	27.80±0.67	30.45±0.93	32.97±0.71
	ESZSL	27.35±0.00	30.32±0.00	37.00±0.00
	ZS-GCN	25.73±0.46	28.62±0.20	37.89±1.15
	ZS-GCN(CNN)	16.01±3.27	21.18±1.58	36.44±0.97
	WDVSc	<u>30.62±0.38</u>	23.46±0.11	38.12±0.35
	Hyperbolic-ZSL	25.36±0.41	34.18±0.88	35.80±2.25
	DGPN (ours)	**34.15±0.28**	**38.16±0.11**	**44.17±0.21**
	Improve↑	+11.53%	+11.61%	+12.68%
Class Split II	RandomGuess	32.69±1.48	50.48±1.70	49.73±1.56
	DAP	30.22±1.21	53.30±0.22	46.79±4.16
	DAP(CNN)	29.83±1.23	50.07±1.70	46.29±0.36
	ESZSL	<u>38.82±0.00</u>	<u>55.32±0.00</u>	<u>56.57±0.00</u>
	ZS-GCN	29.53±0.91	52.22±1.21	55.28±0.41
	ZS-GCN(CNN)	33.20±0.32	49.27±0.73	51.37±1.27
	WDVSc	34.13±0.67	52.70±0.68	46.26±2.58
	Hyperbolic-ZSL	37.02±0.28	46.27±0.39	55.07±0.77
	DGPN (ours)	**48.40±0.31**	**62.40±0.32**	**63.46±0.42**
	Improve↑	+24.68%	+12.80%	+12.18%

The best method is bolded, and the second-best is underlined.

- Our method DGPN always outperforms all baselines by a significant margin, gives 11.94% and 16.55% improvements.
- Baselines still outperform Random Guess.
- Simple classical methods (like DAP and ESZSL) generally get better results than those recently proposed complex ones (like ZS-GCN and Hyperbolic-ZSL)

Compare with Different CSDs

Table 4: Zero-shot node classification accuracy (%) using LABEL-CSDs.

		Cora		Citeseer		C-M10M	
		Acc.	Decl.	Acc.	Decl.	Acc.	Decl.
Class Split I	DAP	25.34	-4.59%	30.01	-12.23%	32.67	-17.5%
	ESZSL	25.79	-5.70%	28.52	-5.94%	35.02	-5.35%
	ZS-GCN	23.73	-7.77%	26.11	-8.77%	33.32	-12.06%
	WDVSc	18.73	-26.14%	19.70	-43.52%	30.82	-13.91%
	Hyperbolic-ZSL	30.47	-3.33%	21.04	-10.32%	34.49	-3.66%
	DGPN (ours)	33.32	-2.34%	31.83	-16.59%	36.05	-31.79%

The results which are better than those of RandomGuess are typeset in blue.
The "Decl." column shows the relative decline, compared to the results in Table 3.

Table 5: Zero-shot node classification accuracy (%) using the graph adjacency information as node attribute information.

		TEXT-CSDs			LABEL-CSDs		
		Cora	Citeseer	C-M10M	Cora	Citeseer	C-M10M
Class Split I	DAP	30.76	33.98	36.76	28.57	19.38	30.91
	ESZSL	24.98	33.20	36.34	30.22	30.05	34.61
	ZS-GCN	28.43	33.35	36.87	23.26	30.26	33.90
	WDVSc	18.98	28.77	33.84	29.73	23.03	30.35
	Hyperbolic-ZSL	19.96	12.16	35.80	28.53	12.45	30.82
	DGPN (ours)	32.96	38.03	40.01	31.28	31.85	35.75

The results which are better than those of RandomGuess are typeset in blue.

The performance of all methods (including ours) declines significantly, compared to those results in Table 3 where TEXT-CSDs are used.

This indicates node attributes contain richer and useful information than graph structure information.

Compare Node Attributes with Adjacency Matrix

Table 5: Zero-shot node classification accuracy (%) using the graph adjacency information as node attribute information.

		TEXT-CSDs			LABEL-CSDs		
		Cora	Citeseer	C-M10M	Cora	Citeseer	C-M10M
Class Split I	DAP	30.76	33.98	36.76	28.57	19.38	30.91
	ESZSL	24.98	33.20	36.34	30.22	30.05	34.61
	ZS-GCN	28.43	33.35	36.87	23.26	30.26	33.90
	WDVSc	18.98	28.77	33.84	29.73	23.03	30.35
	Hyperbolic-ZSL	19.96	12.16	35.80	28.53	12.45	30.82
	DGPN (ours)	32.96	38.03	40.01	31.28	31.85	35.75

The results which are better than those of RandomGuess are typeset in blue.

This indicates node attributes contain richer and useful information than graph structure information.

Model Ablation

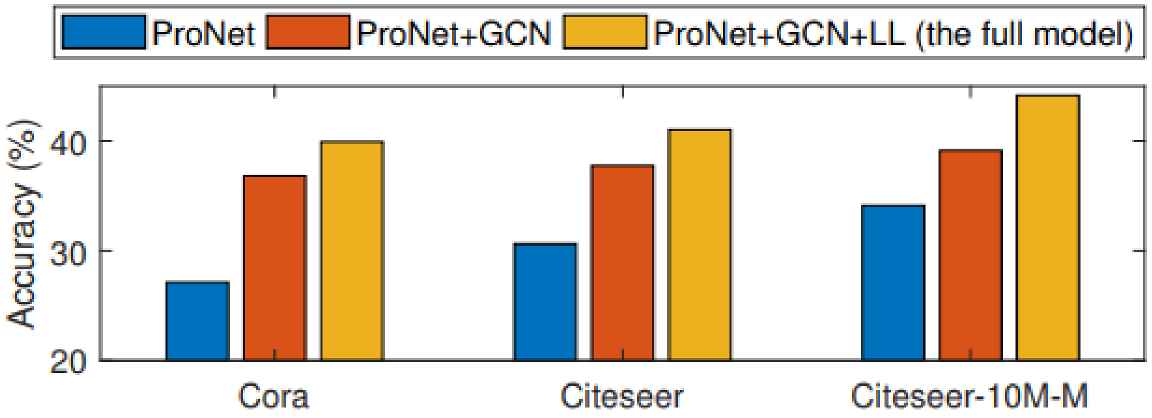
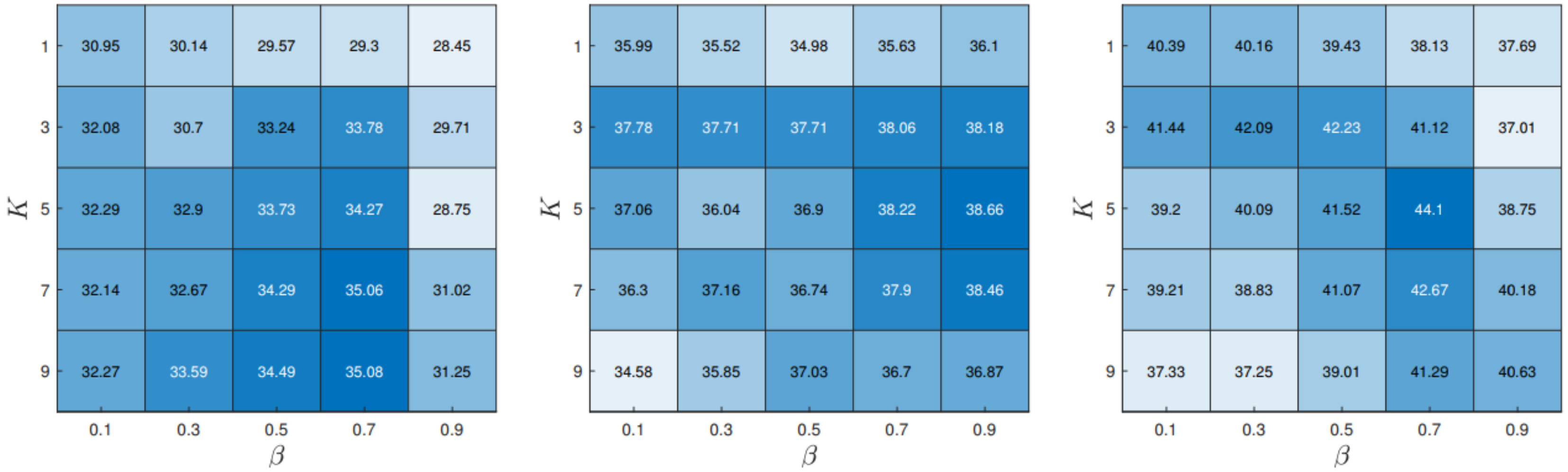


Figure 3: Model ablation under Class Split I.

- ProNet: the variant that replaces the decomposed GCNs part with fully-connect layer
- ProNet+GCN: the variant that removes the local loss part in our method.
- ProNet+GCN+LL: full model

Both two parts (the decomposed GCNs part and local loss part) contribute to the final performance, which evidently demonstrates their effectiveness.

Hyper-Parameters Searching



(a) Cora (b) Citeseer (c) C-M10M

Figure 4: Effects of K and β in our method under Class Split I. Grid numbers denote the classification accuracy (%). Color indicates the performance (the deeper the better).

- Show the usefulness of the graph structure information and the lazy random walk strategy.

Some interesting findings

1. The quality of CSDs is the key to the ZNC problem; we can rank the importance as: CSDs ≫node attributes>graph structure.
2. Comparing to “RandomGuess”, we can rank the general performance of those two CSDs as: TEXT-CSDs ≫ LABEL-CSDs ⩾ RandomGuess.
3. With high-quality CSDs, graph structure information can be very useful or even be comparable to node attributes.
4. Through subtly recasting the concepts, locality and compositionality can be well adapted to graph-structured data.

Summary

- Three main contributions:
  - Novel problem: Zero-shot Node Classification (ZNC)
  - Novel CSDs acquisition and evaluation strategy
  - Novel zero-shot method DGPN
- Code available at: https://github.com/zhengwang100/dgpn
- Other related topics
  - Node Classification: https://en.wikipedia.org/wiki/Collective_classification
  - Zero-shot Learning (ZSL): https://en.wikipedia.org/wiki/Zero-shot_learning
  - Zero-shot Graph Embedding (ZGE): https://zhengwang100.github.io/project/zero_shot_graph_embedding.html

THANK YOU!

Zheng Wang https://zhengwang100.github.io