SMU SINGAPORE MANAGEMENT UNIVERSITY
School of Computing and Information Systems
salesforce

Relative and Absolute Location Embedding for Few-Shot Node Classification on Graph

Zemin Liu, Yuan Fang, Chenghao Liu, Steven C.H. Hoi

Motivation

Problem

Few-shot node classification for novel classes

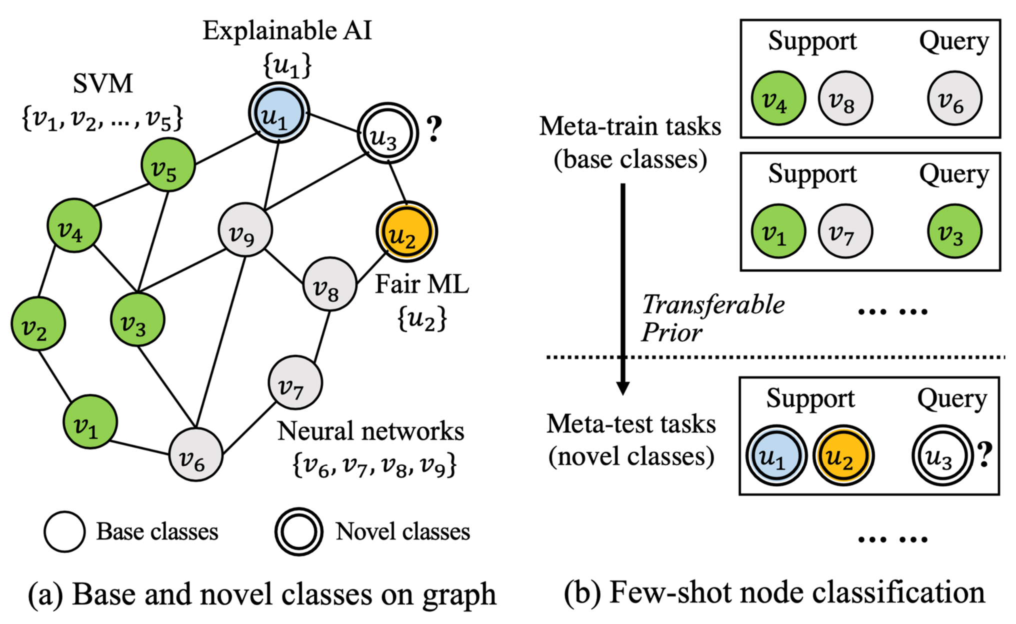

Figure 1: Illustration of few-shot node classification.

What's missing in SOTA?

- Graph nodes related to each other and the inter-dependency is crucial to learning node representations.
- SOTA models [1, 2] overlook such inter-dependency in a task, which should be incorporated into the transferable prior.

Challenges

- How to capture the potentially long-ranged dependencies between nodes within a task?
- How to align the dependencies across tasks to converge on a common transferable prior?

The proposed model: RALE

Overall framework

Key concept: Hub nodes

- Structurally important nodes [3]
- Utilize hubs for efficient sampling of important paths for *relative location*
- Utilize hubs as a set of global references, to align the dependencies across tasks for *absolute location*

Abstraction of Location Embedding

- *Graph encoder* ϕ_g: embed nodes
- *Path encoder* ϕ_p: embed paths
- *Location embedding w.r.t.* reference nodes $\mathcal{R}$

$$\mathbf{e}_v^{\{u\}} = \text{AGGR}(\{\phi_p(\phi_g(p;\theta_g);\theta_p) : \forall p \in \mathcal{P}_{u,v}\})$$

$$\mathbf{e}_v^{\mathcal{R}} = \phi(\{\mathcal{P}_{u,v} : u \in \mathcal{R}\}; \theta_g, \theta_p) = \text{AGGR}(\{\mathbf{e}_v^{\{u\}} : \forall u \in \mathcal{R}\}).$$

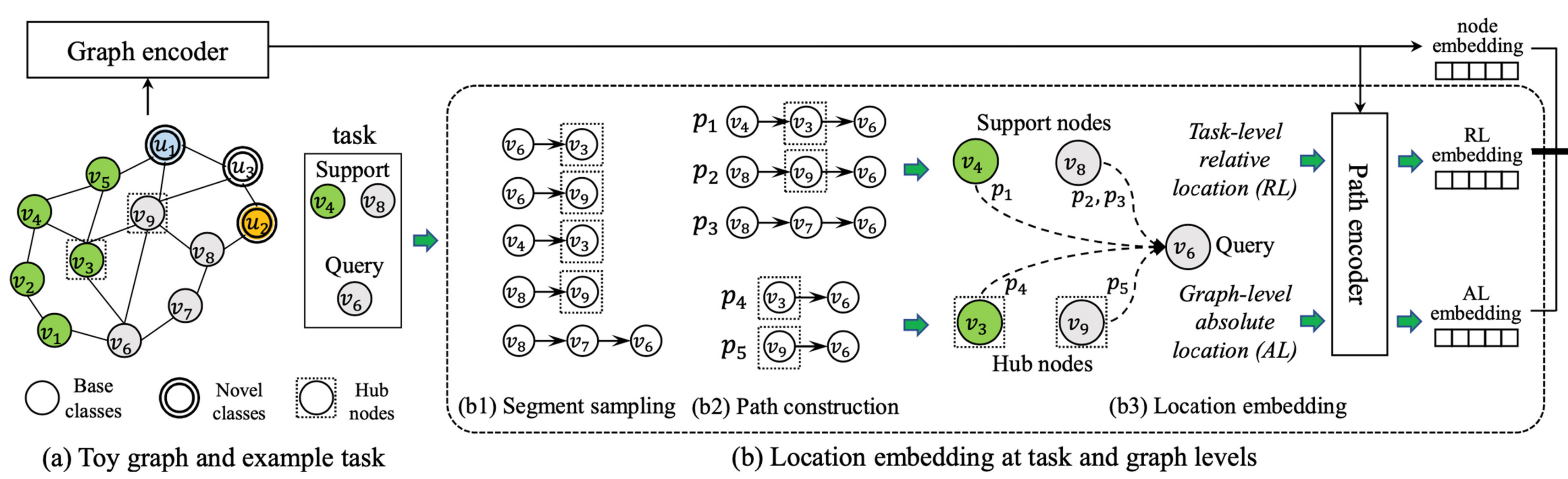

Figure 3: Overview of the proposed model RALE.

Task- and graph-level location embedding

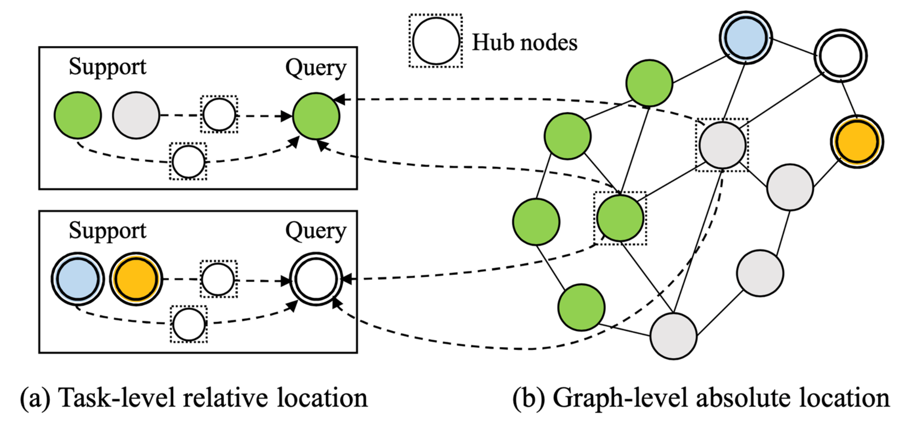

Figure 2: Hub-based relative and absolute locations.

- Task-level relative location
  - Task $t = (S_t, Q_t)$

$$\mathbf{e}_v^{S_t} = \phi(\{\mathcal{P}_{s,v} : s \in S_t\}; \theta_g, \theta_p)$$

- Graph-level absolute location
  - Hub set $\mathcal{H}$

$$\mathbf{e}_v^{\mathcal{H}} = \phi(\{\mathcal{P}_{h,v} : h \in \mathcal{H}\}; \theta_g, \theta_p)$$

Dependency-aware meta-optimization

- Dependency-aware classification layer

$$\psi(v;\Theta) = \textsc{Softmax}\left(\sigma(\mathbf{W}\left[\mathbf{h}_v \| \mathbf{e}_v^{S_t} \| \mathbf{e}_v^{\mathcal{H}}\right])\right)$$

node emb. relative location emb. absolute location emb.

- Meta-objective (MAML [4])

$$L(S_t;\Theta) = -\sum_{v\in S_t}\sum_{i=1}^{m} \mathbb{I}_{\ell(v)=i} \ln(\psi(v;\Theta)[i])$$ (Loss on the support)

$$\Theta' = \Theta - \alpha\frac{\partial L(S_t;\Theta)}{\partial\Theta}$$ (Adapting prior Θ to Θ' by one gradient update)

$$\Theta^* = \arg\min_\Theta \sum_{t\in\mathcal{T}_{tr}} L(Q_t; \Theta - \alpha\frac{\partial L(S_t;\Theta)}{\partial\Theta})$$ (Optimization loss on the query)

Experiments

Datasets

Table 1: Summary of datasets.

	Nodes	Edges	Features	Classes (Train/Val/Test)
Amazon	13,381	245,778	767	10 (5/2/3)
Email	909	13,733	128	28 (15/6/7)
Reddit	231,371	11,606,876	602	41 (25/6/10)

Baselines

GNNs: GCN [5], GraphSAGE [6], GAT [7]

- Training: optimize GNN on all base classes
- Test: freeze parameters of aggregation layer, update parameters of classification layer

GNN+'s: GCN+, GraphSAGE+, GAT+

- Training: optimize GNN on all base classes
- Test: update all parameters

Meta-learning models: Meta-GNN, Proto-GNN

- Meta-GNN: MAML [4] based model
- Proto-GNN: prototypical network [8] based model

Main results

Table 2: Accuracy (percent) of RALE and baselines. In each row, the best result is bolded and the second best is underlined. RALE's improvement is calculated relative to the best baseline, with the corresponding p-value under two-tail paired t-test.

		GCN	GraphSAGE	GAT	GCN+	GraphSAGE+	GAT+	Meta-GNN	Proto-GNN	RALE	impr.	p-value
Amazon (2-way)	1-shot	71.97	63.19	72.61	70.86	68.36	66.99	<u>73.20</u>	68.91	**78.07**	+6.65%	0.007
	3-shot	<u>78.67</u>	63.15	78.18	72.66	69.75	76.07	74.45	76.41	**84.17**	+6.99%	0.021
	5-shot	79.46	64.71	<u>85.18</u>	79.84	68.44	**85.45**	76.58	79.67	84.50	-1.11%	0.437
Email (5-way)	1-shot	43.15	40.25	43.11	47.15	43.48	40.82	<u>47.21</u>	35.47	**51.82**	+9.76%	< 0.001
	3-shot	53.29	45.56	41.06	50.51	44.28	35.83	<u>55.64</u>	40.93	**59.47**	+6.88%	0.002
	5-shot	58.19	48.38	37.24	58.37	47.01	33.52	<u>58.76</u>	42.38	**65.46**	+11.40%	< 0.001
Reddit (5-way)	1-shot	20.02	41.61	19.95	20.13	35.89	19.99	<u>46.42</u>	44.95	**48.63**	+4.76%	< 0.001
	3-shot	20.16	47.00	20.15	20.21	49.11	20.02	51.28	<u>51.60</u>	**56.85**	+10.17%	< 0.001
	5-shot	20.40	50.53	20.12	20.41	51.30	20.14	<u>53.33</u>	52.57	**57.45**	+7.73%	0.029

Ablation study

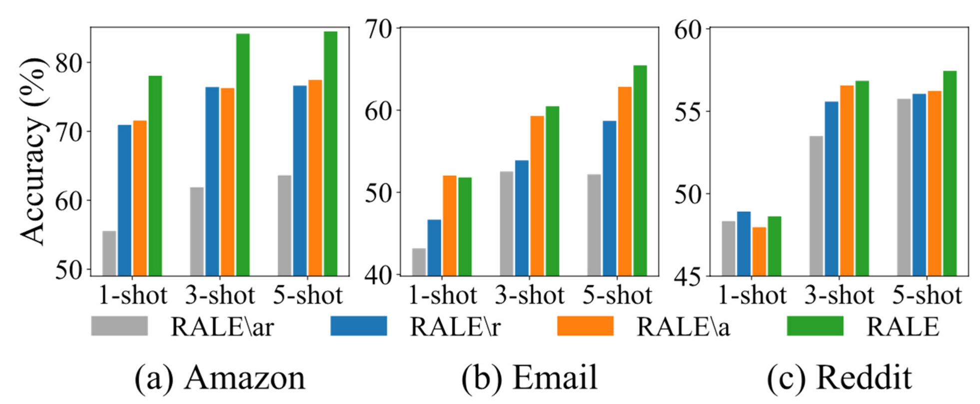

Figure 4: Ablation study.

- RALE\a: no absolute location embedding
- RALE\r: no relative location embedding
- RALE\ar: without both location embeddings
- RALE: the full model

Conclusion

- **Problem**
  - Few-shot node classification for novel classes on graph
- **Motivation**
  - Nodes on a graph are non-i.i.d instances; they relate to each other
  - Need to capture the node dependencies within and across tasks
- **Proposed model: RALE**
  - Hub-based relative and absolute location embedding
  - Relative locations capture the task-level dependency
  - Absolute locations capture the graph-level dependency to align the tasks

References

[1] Zhou, F., et al. 2019. Meta-GNN: On few-shot node classification in graph meta-learning. In *CIKM*.

[2] Yao, H., et al. 2020. Graph few-shot learning via knowledge transfer. In *AAAI*.

[3] Page, L., et al. 1999. The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*.

[4] Finn C, et al. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.