Jianhao GUO, Zixuan NI, Yun ZHU, Siliang TANG, 2025. E-CGL: an efficient continual graph learner. *Frontiers of Information Technology & Electronic Engineering*, 26(8):1441-1453. https://doi.org/10.1631/FITEE.2500162

E-CGL: an efficient continual graph learner

Key words: Graph neural networks; Continual learning; Dynamic graphs; Continual graph learning; Graph acceleration

Corresponding author: Siliang TANG
E-mail: siliang@zju.edu.cn
ORCID: https://orcid.org/0000-0002-7356-9711

Motivation

1. The first challenge arises from the topological interdependence of graph snapshots, in which new tasks inherit structural and attributive dependencies from previous tasks. Topological interdependence occurs when sequential graph snapshots $\{G_1, G_2, ..., G_T\}$ share nodes/edges or have attribute continuity, resulting in conditional dependencies between their distributions.

2. The second challenge is the efficiency and scalability of handling increasingly large graphs. Real-world graphs (e.g., social networks) grow exponentially, necessitating efficient model updates.

Main contributions

1. Interdependence solution. We develop graph-aware replay strategies with importance and diversity sampling to explicitly capture inter-task dependencies. This strategies preserves critical information across updates.

2. Efficiency solution. We design an efficient graph learner with a shared-weight multilayer perceptron (MLP) encoder that accelerates training by up to 28.4$\times$ and reintroduces structural information during inference with the minimal performance compromises.

3. Empirical validation. We validate E-CGL on four datasets for supervised node classification using both task-incremental (task-IL) and class-incremental (class-IL) settings, achieving state-of-the-art performance in accuracy, forgetting, and computational efficiency.

Method

1. The interdependencies between historical data and current data make continual graph learning challenging. By applying Bayes' rule to the prior probability of the parameters $p(\theta)$ and likelihood function $p(D|\theta)$, we prove the necessity of the replay-based strategy from a probabilistic perspective.

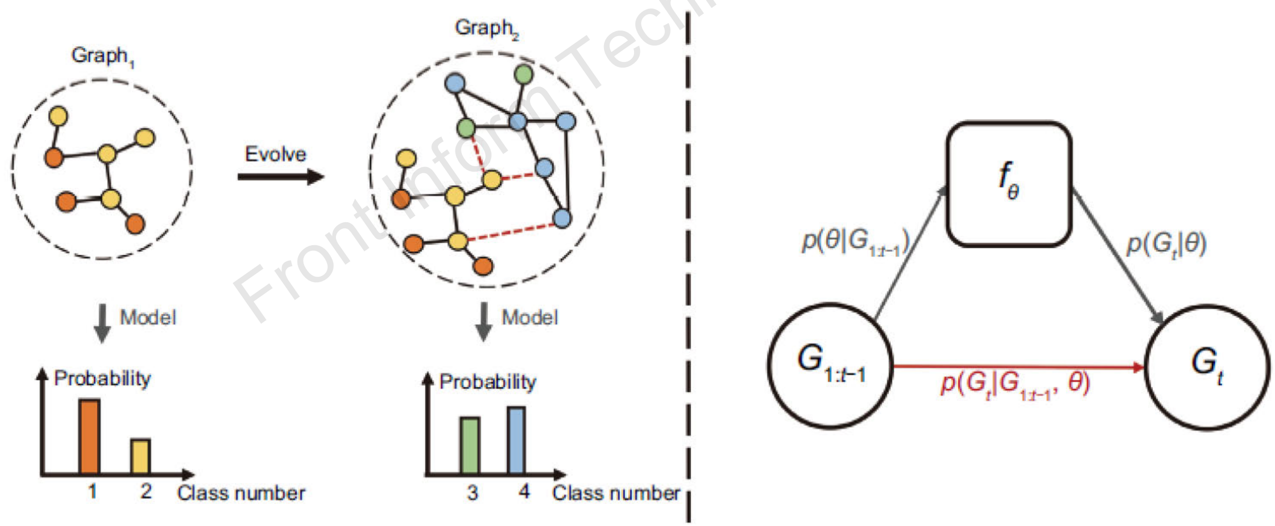

Fig. 1 Left: visualization of continual graph learning (CGL). Right: illustration of conditional probabilities for CGL. The gray lines show Bayes' rule for independent identically distributed data. The red line represents the influence of previous data on the current graph. References to color refer to the online version of this figure

Method (Cont'd)

2. Two heuristic sampling strategies are proposed for graph replay: importance-based sampling and diversity-based sampling. The former leverages the attributed PageRank algorithm to calculate the importance of each node, whereas the latter measures the diversity of a node based on its representation divergence compared to its neighbors.

$$\boldsymbol{\pi}_{\mathrm{Imp}} = d\boldsymbol{T}\boldsymbol{\pi}_{\mathrm{Imp}} + (1-d)\boldsymbol{r}.$$

$$\pi_{\mathrm{Div}}(i) = \left\|\boldsymbol{x}_i - \frac{1}{|\mathcal{N}_i|}\sum_{u\in\mathcal{N}_i}\boldsymbol{x}_u\right\|.$$

$$\mathcal{M} = \mathcal{M} \cup \mathrm{argtopk}(\boldsymbol{\pi}_{\mathrm{Imp}}) \cup \mathrm{argtopk}(\boldsymbol{\pi}_{\mathrm{Div}}),$$

Method (Cont'd)

3. To improve the training efficiency for large graph updates, a graph convolution network (GCN) is decomposed into two parts: the message passing module and the feature transformation module. The time-consuming message passing module is removed during training, so the GCN is reduced to an MLP. For inference, the message passing module is added back to maintain performance.

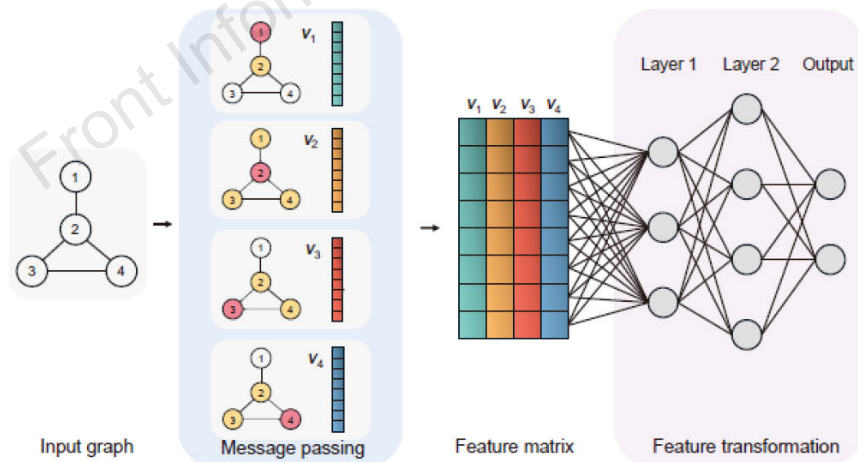

Fig. 2 Architecture of GCN, which consists of the message passing and feature transformation processes. Our E-CGL removes the message passing process during training, simplifying the training process to an MLP network

Major results

Task-IL results

Table 1 Results on node classification tasks under task-IL setting

Category	Method	CoraFull		OGBN-Arxiv		Reddit		OGBN-Products	
		AA (%)	AF (%)	AA (%)	AF (%)	AA (%)	AF (%)	AA (%)	AF (%)
Lower-bound	Fine-tuning	39.5±1.8	−54.6±1.7	51.5±4.6	−34.7±5.0	57.9±3.5	−44.9±3.9	68.7±2.0	−27.1±2.4
Upper-bound	Joint training	91.0±0.2	N/A	83.8±0.5	N/A	98.2±0.1	N/A	94.4±0.3	N/A
CV-based	LwF	56.7±5.9	−35.2±6.4	79.2±0.7	−2.1±0.8	72.6±6.4	−28.1±7.2	73.8±2.0	−22.4±1.8
	EWC	66.4±3.3	−24.3±3.6	62.3±4.3	−16.3±4.5	87.5±4.4	−11.3±5.4	92.3±1.4	−1.0±0.3
	MAS	86.4±0.8	−0.2±0.3	81.0±0.8	0.0±0.0	89.7±1.7	**0.0**±0.2	83.6±1.0	**−0.1**±0.1
	GEM	81.1±0.8	−8.8±1.1	67.9±0.4	−9.3±0.6	55.6±23.3	−4.9±3.9	78.8±15.3	−1.7±3.0
Graph-based	TWP	86.4±0.4	−1.8±0.7	80.5±0.8	−1.7±0.8	87.8±5.5	−11.5±6.1	93.1±1.5	−1.2±0.9
	ER-GNN	88.9±0.1	**0.1**±0.3	75.0±0.3	−8.2±0.5	89.8±0.6	−9.2±0.8	93.3±0.7	−0.9±0.1
	DyGRAIN	N/A	N/A	70.9±0.3	−4.6±0.1	92.1±0.5	−3.5±0.1	73.4±0.2	−3.3±0.1
	SSM	88.9±0.5	−0.8±0.1	80.8±1.3	−2.4±0.7	**93.7**±0.3	−4.9±0.6	91.6±0.6	−3.6±0.2
	CaT	84.3±0.5	−6.1±0.8	70.6±4.2	−14.2±2.3	90.9±0.3	−5.5±0.5	OOM	OOM
	GCL-SAGE	87.1±0.7	−4.5±0.2	77.9±1.2	−12.4±4.0	91.0±0.4	−5.8±0.6	89.5±0.5	−3.9±0.4
	TACO	85.2±0.6	−6.6±0.9	79.3±1.5	−15.3±1.2	89.5±0.7	−7.0±1.8	87.2±0.5	-4.9±0.6
	DSLR	89.1±0.2	−1.5±0.3	80.4±0.9	−3.2±0.5	91.9±0.4	−2.3±0.4	92.7±0.3	−1.5±0.2
Ours	E-CGL	**89.6**±0.1	−2.5±0.2	**82.1**±1.0	**0.2**±0.2	92.2±0.7	−2.7±0.8	**93.9**±0.6	−1.2±0.3

For joint training, it is trained on all tasks as the upper bound; therefore, we do not provide its AF values. DyGRAIN has no open-source code, and thus we use the results reported in the reference (Kim et al., 2022). Negative values for AF indicate performance drop due to catastrophic forgetting, and N/A refers to not available. The best results are marked in bold, and the second-best are underlined

Major results (Cont'd)

Class-IL results

Table 2 Results on node classification tasks under class-IL setting

Category	Method	CoraFull		OGBN-Arxiv		Reddit		OGBN-Products	
		AA (%)	AF (%)	AA (%)	AF (%)	AA (%)	AF (%)	AA (%)	AF (%)
Lower-bound	Fine-tuning	5.5±0.0	−90.7±0.3	8.5±0.3	−80.9±0.5	11.6±1.3	−93.6±2.5	4.4±1.4	−83.7±3.5
Upper-bound	Joint training	79.5±0.1	N/A	51.2±0.2	N/A	97.2±0.5	N/A	76.3±0.2	N/A
CV-based	LwF	5.9±0.5	−91.2±0.8	8.8±0.2	−81.3±0.5	11.8±0.4	−89.8±2.6	5.4±0.1	−88.7±1.1
	EWC	5.4±0.1	−91.1±3.4	8.8±0.2	−83.1±1.4	11.9±2.1	−95.3±1.7	3.5±0.3	−91.8±0.9
	MAS	4.3±0.1	−86.9±1.1	8.7±0.0	−77.0±0.7	9.1±0.2	−51.0±2.5	19.1±0.7	−16.2±0.1
	GEM	6.4±0.6	−89.9±0.9	8.8±0.1	−80.5±0.5	9.2±1.3	−11.2±4.8	7.7±3.0	−**15.3**±2.6
Graph-based	TWP	7.5±0.1	−86.3±0.5	8.5±0.1	−81.4±0.5	13.4±1.3	−93.6±1.4	3.3±0.6	−94.8±0.3
	ER-GNN	5.1±0.2	−90.7±0.2	17.3±0.8	−69.6±1.1	62.0±4.5	−39.4±5.1	6.8±1.6	−90.2±2.2
	SSM	**76.2**±1.2	**6.0**±0.1	33.0±0.4	**21.5**±0.4	74.8±1.7	−**7.1**±0.7	OOM	OOM
	CaT	14.8±5.2	−81.3±3.7	32.1±0.3	−48.8±0.5	74.1±2.2	−20.8±1.2	OOM	OOM
	GCL-SAGE	34.2±3.0	−17.7±5.3	23.5±1.8	−42.3±3.2	65.3±2.1	−25.1±2.5	34.8±1.5	−41.9±1.8
	TACO	71.5±1.3	−12.5±1.1	31.2±0.8	−35.6±0.9	73.2±1.5	−18.9±1.7	51.4±0.7	−28.3±0.6
	DSLR	75.8±0.7	−8.2±0.4	**34.5**±0.6	−28.1±0.5	77.9±0.8	−9.5±0.3	54.7±0.9	−19.8±0.4
Ours	E-CGL	68.2±0.2	−17.8±0.2	33.7±0.2	−41.0±0.1	**78.6**±1.0	−17.3±1.2	**56.3**±1.0	−31.0±1.1

For joint training, it is trained on all tasks as the upper bound; therefore, we do not provide its AF values. The results of DyGRAIN in the class-IL setting are not provided in the reference. Negative values for AF indicate performance drop due to catastrophic forgetting, and N/A refers to not available. The best results are marked in bold, and the second-best are underlined

Major results (Cont'd)

Runtime analysis

Table 3 Runtime (training/inference) comparison of different CGL methods under task-IL setting

Method	Runtime (ms)				Average time (ms)
	CoraFull	OGBN-Arxiv	Reddit	OGBN-Products	
TWP	34.25/105.71	37.33/23.49	354.68/1104.26	3217.46/2894.05	910.93/1031.88
ER-GNN	56.51/107.67	47.17/22.64	421.83/1075.20	4419.80/4385.23	1236.33/1397.69
SSM	52.68/109.89	49.63/27.65	491.99/1220.88	3976.74/4351.92	1142.76/1427.59
CaT	176.24/115.81	152.13/23.15	1719.85/1120.00	OOM	N/A
E-CGL (GCN)	28.94/110.79	43.18/27.39	1078.31/1032.93	5205.96/4912.76	1589.10/1520.97
E-CGL	**15.69/99.24**	**25.61/17.95**	**177.13/574.13**	**183.04/552.72**	**100.37/311.01**
Improvement	1.84×/1.12×	1.69×/1.53×	6.09×/1.80×	28.44×/8.89×	15.83×/4.89×

E-CGL (GCN) is the version that we use GCN as the encoder for both training and inference. Improvement indicates the performance improvement of our E-CGL compared with E-CGL (GCN). We report both training and inference time. The best results are marked in bold

Major results (Cont'd)

Memory usage analysis

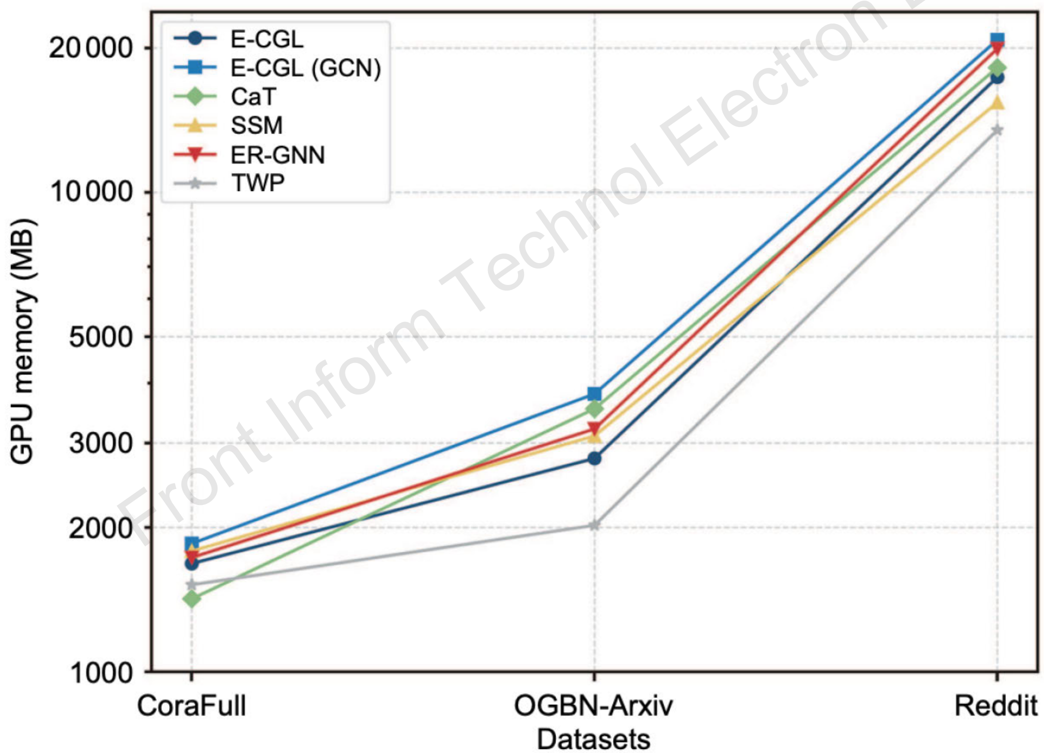

Fig. 3 GPU memory usage of different CGL methods under the task-IL setting

Conclusions

1. With the topology-aware replay strategy, a continual graph learning algorithm was proposed to alleviate catastrophic forgetting during graph updates.

2. With an MLP encoder that shares weights with its counterpart GCN, the algorithm supported training acceleration by up to 28 times.

3. Extensive empirical results demonstrated the effectiveness of the proposed method in terms of both performance and efficiency.

Jianhao Guo received his B.E. degree from Zhejiang University, Hangzhou, China, in 2020. He is currently working toward the Ph.D. degree in computer science in Zhejiang University, under the supervision of Prof. Siliang TANG.

Siliang TANG is currently a full professor at the College of Computer Science, Zhejiang University. He received his Ph.D. degree from the National University of Ireland, Maynooth, Ireland. His research interests include natural language processing, cross-media analysis, and graph mining. So far, he has published more than a hundred papers in the top-tier scientific conferences/journals such as AAAI, IJCAI, KDD, NIPS, ICML, ACL, EMNLP, NAACL, SIGIR, IEEE T-KDE (NLP and graph mining), ACM MM, CVPR, ICCV, IEEE T-MM, IEEE T-IP, IEEE T-CSVT (multimodal understanding, and graph neural networks in computer vision), IEEE VIS, and IEEE T-VCG (data visualization). He has been serving as an area chair/senior PC member or PC member in conferences such as AAAI, IJCAI, NIPS, ICML, CVPR, ACL, EMNLP, and NAACL, and reviewers of journals such as *IEEE TIP*, *IEEE TMM*, *IEEE TSMC*, *ACM Comput Surv*, and *Sci Rep*.