Replay-and-Forget-Free Graph Class-Incremental Learning: A Task Profiling and Prompting Approach

Chaoxi Niu[1], Guansong Pang[2], Ling Chen[1], Bing Liu[3]

[1] AAII, University of Technology Sydney, Australia

[2] Singapore Management University, Singapore

[3] University of Illinois at Chicago, USA

UTS CRICOS 00099F

Framework

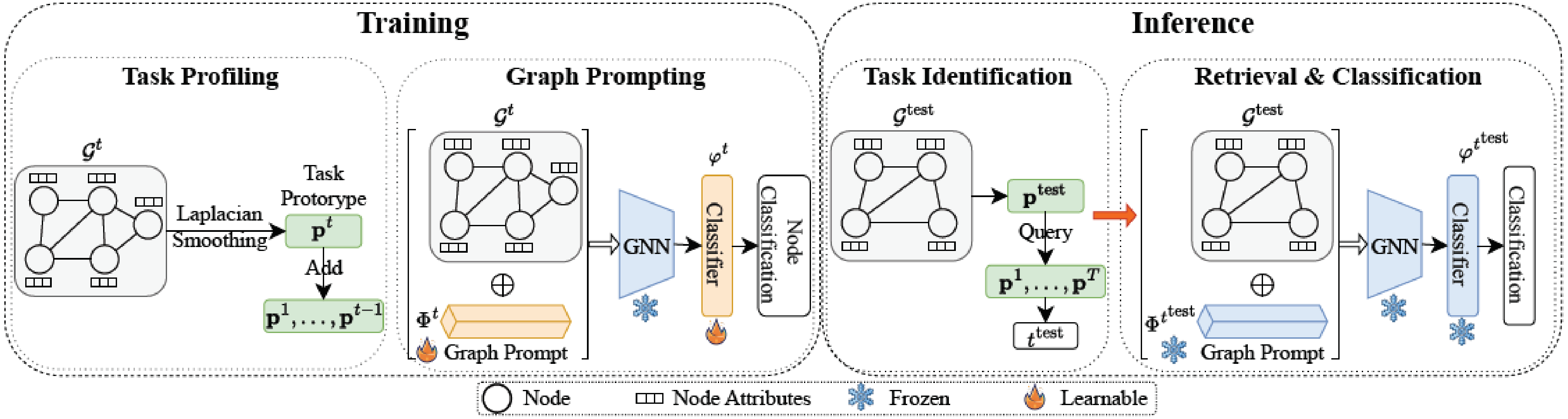
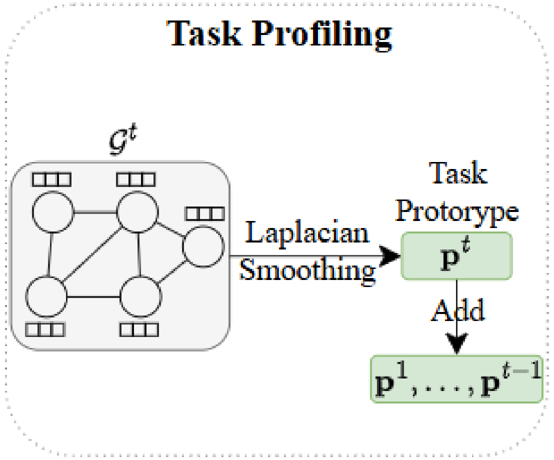


Figure 2: Overview of the proposed TPP approach. During training, for each graph task t, the task prototype $\mathbf{p}^t$ is generated by applying Laplacian smoothing on the graph $\mathcal{G}^t$ and added to $\mathcal{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^{t-1}\}$. At the same time, the graph prompt Φ^t and the classification head φ^t for this task are optimized on $\mathcal{G}^t$ through a frozen pre-trained GNN. During inference, the task ID of the test graph is first inferred (*i.e.*, task identification). Then, the graph prompt and the classifier of the predicted task are retrieved to perform the node classification in GCIL. The GNN is trained on $\mathcal{G}^1$ and remains frozen for subsequent tasks.

Method

Task Profiling for Task ID Prediction



Graph Laplacian Smoothing

$$Z^t = (I - (\hat{D}^t)^{-\frac{1}{2}} \hat{L}^t (\hat{D}^t)^{-\frac{1}{2}})^s X^t$$

Task Prototype

$$\mathbf{p}^t = \frac{1}{|\mathcal{V}^t_{\text{train}}|} \sum_{i \in \mathcal{V}^t_{\text{train}}} \mathbf{z}^t_i (\hat{D}^t_{ii})^{-\frac{1}{2}}$$

Test Task Prototype

$$\mathbf{p}^{\text{test}} = \frac{1}{|\mathcal{V}^{\text{test}}|} \sum_{i \in \mathcal{V}^{\text{test}}} \mathbf{z}^{\text{test}}_i (\hat{D}^{\text{test}}_{ii})^{-\frac{1}{2}}$$

Task Inference

$$t^{\text{test}} = \arg\min(d(\mathbf{p}^{\text{test}}, \mathbf{p}^1), \ldots, d(\mathbf{p}^{\text{test}}, \mathbf{p}^T)))$$

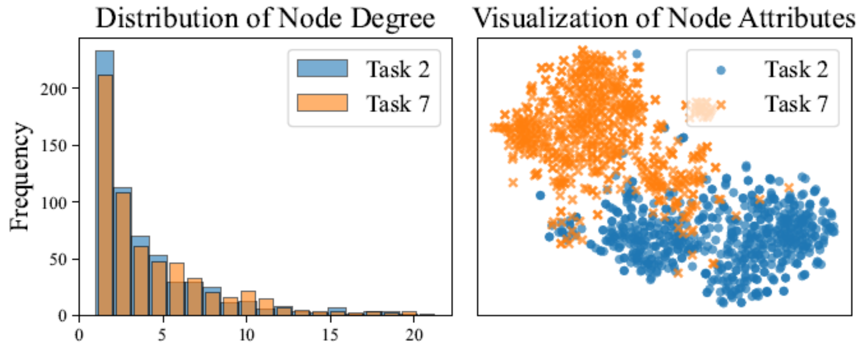
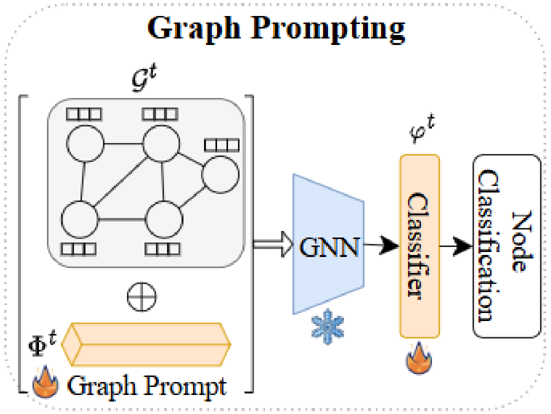


Figure 3: The differences between two graphs in structure and node attributes.

Method

Graph Prompt Learning



Prompt Design

$$\bar{\mathbf{x}}_i^t = \mathbf{x}_i^t + \sum_j^k \alpha_j \phi_j^t \,, \quad \alpha_j = \frac{e^{(\mathbf{w}_j)^T \mathbf{x}_i^t}}{\sum_l^k e^{(\mathbf{w}_l)^T \mathbf{x}_i^t}}$$

Node Classification with Prompt

$$\hat{Y}^t = \varphi^t(f(A^t, X^t + \Phi^t))$$

Training Objective

$$\min_{\Phi^t, \varphi^t} \frac{1}{|\mathcal{V}^t_{\text{train}}|} \sum_{i \in \mathcal{V}^t_{\text{train}}} \ell_{\text{CE}}(\hat{y}_i^t, y_i^t)$$

Results

Table 1: Results (mean±std) under the GCIL setting on four large datasets. The best performance on each dataset is boldfaced. "↑" denotes the higher value represents better performance. Oracle Model can get access to the data of all tasks and task IDs, *i.e.*, it obtains the upper bound performance. "✓" in Data Replay indicates the use of data replay in the model, and × denotes no data replay involved.

Methods	Data Replay	CoraFull		Arixv		Reddit		Products	
		AA/%↑	AF/%↑	AA/%↑	AF/%↑	AA/%↑	AF/%↑	AA/%↑	AF/%↑
Fine-tune	×	3.5±0.5	-95.2±0.5	4.9±0.0	-89.7±0.4	5.9±1.2	-97.9±3.3	7.6±0.7	-88.7±0.8
Joint	×	81.2±0.4	-	51.3±0.5	-	97.1±0.1	-	71.5±0.1	-
EWC	×	52.6±8.2	-38.5±12.1	8.5±1.0	-69.5±8.0	10.3±11.6	-33.2±26.1	23.8±3.8	-21.7±7.5
MAS	×	6.5±1.5	-92.3±1.5	4.8±0.4	-72.2±4.1	9.2±14.5	-23.1±28.2	16.7±4.8	-57.0±31.9
GEM	×	8.4±1.1	-88.4±1.4	4.9±0.0	-89.8±0.3	11.5±5.5	-92.4±5.9	4.5±1.3	-94.7±0.4
LwF	×	33.4±1.6	-59.6±2.2	9.9±12.1	-43.6±11.9	86.6±1.1	-9.2±1.1	48.2±1.6	-18.6±1.6
TWP	×	62.6±2.2	-30.6±4.3	6.7±1.5	-50.6±13.2	8.0±5.2	-18.8±9.0	14.1±4.0	-11.4±2.0
ERGNN	✓	34.5±4.4	-61.6±4.3	21.5±5.4	-70.0±5.5	82.7±0.4	-17.3±0.4	48.3±1.2	-45.7±1.3
SSM-uniform	✓	73.0±0.3	-14.8±0.5	47.1±0.5	-11.7±1.5	94.3±0.1	-1.4±0.1	62.0±1.6	-9.9±1.3
SSM-degree	✓	75.4±0.1	-9.7±0.0	48.3±0.5	-10.7±0.3	94.4±0.0	-1.3±0.0	63.3±0.1	-9.6±0.3
SEM-curvature	✓	77.7±0.8	-10.0±1.2	49.9±0.6	-8.4±1.3	96.3±0.1	-0.6±0.1	65.1±1.0	-9.5±0.8
CaT	✓	80.4±0.5	-5.3±0.4	48.2±0.4	-12.6±0.7	97.3±0.1	-0.4±0.0	70.3±0.9	-4.5±0.8
DeLoMe	✓	81.0±0.2	-3.3±0.3	50.6±0.3	5.1±0.4	97.4±0.1	-0.1±0.1	67.5±0.7	-17.3±0.3
OODCIL	✓	71.3±0.5	-1.1±0.1	19.3±1.4	-1.0±0.4	79.3±0.8	-0.1±0.0	41.6±0.9	-1.6±0.4
TPP (Ours)	×	**93.4±0.4**	**0.0±0.0**	**85.4±0.1**	**0.0±0.0**	**99.5±0.0**	**0.0±0.0**	**94.0±0.5**	**0.0±0.0**
Oracle Model	×	95.5±0.2	-	90.3±0.4	-	99.5±0.0	-	95.3±0.8	-