Node-Time Conditional Prompt Learning in Dynamic Graphs

Xingtong Yu, Zhenghao Liu, Xinming Zhang, Yuan Fang

SMU SINGAPORE MANAGEMENT UNIVERSITY

University of Science and Technology of China

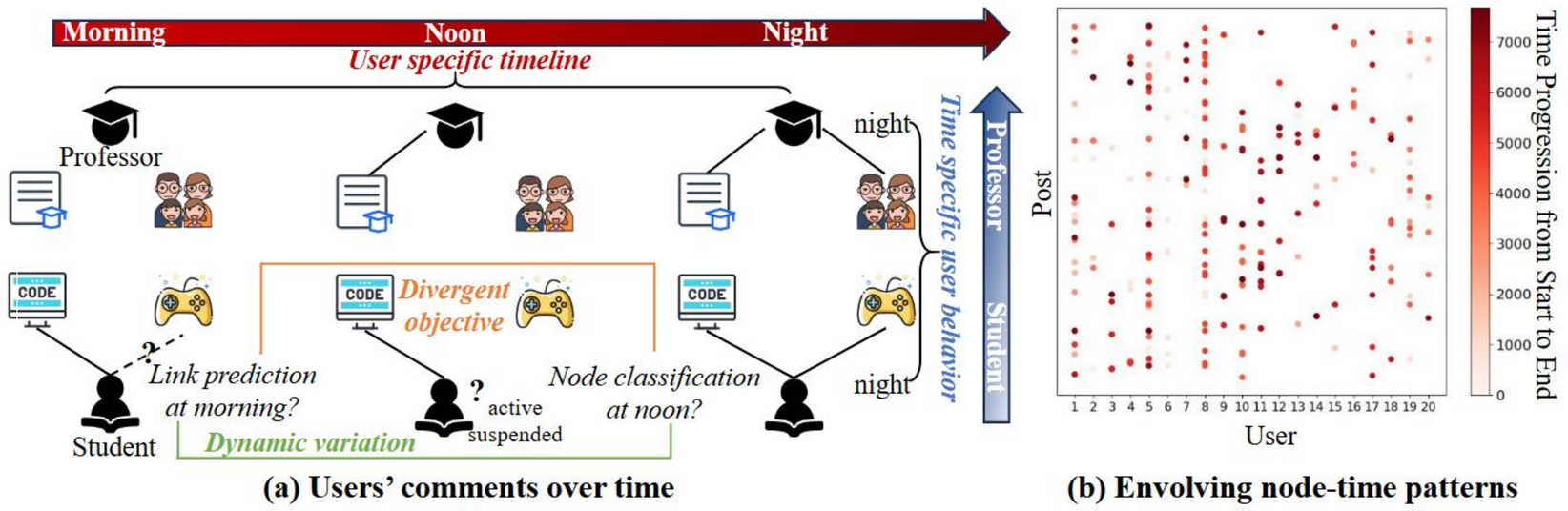
Motivation

Problem

- Design a prompt learning framework for dynamic graphs

Challenges

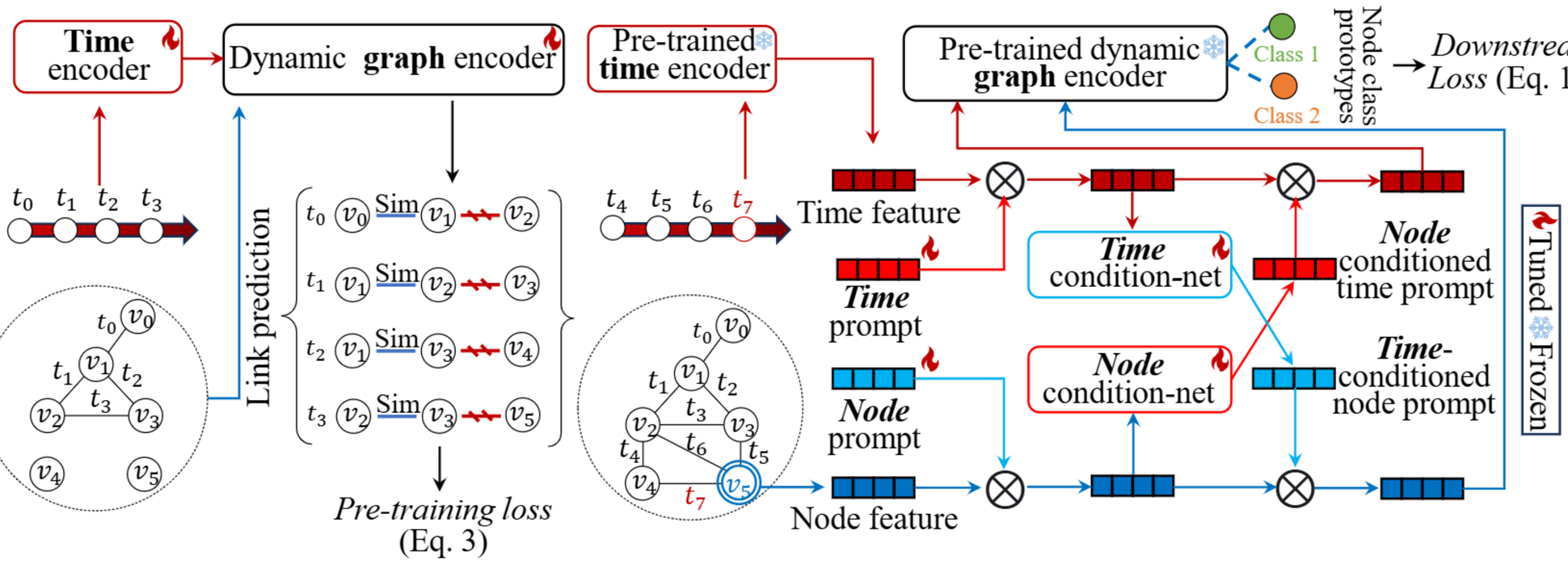
- *How do we design prompts to bridge temporal variations across time, and divergent task?*
- *How do we capture evolving patterns across different nodes and time points?*



(a) Users' comments over time (b) Envolving node-time patterns

Figure 1: Motivation of DyGPrompt. (a) Users comment on different topics over time. (b) Evolving node-time patterns in Wikipedia, as node and time mutually characterize each other.

Proposed Method: DyGPrompt



(a) Toy dynamic graph (b) Pre-training (c) Prompt tuning for downstream tasks

Figure 2: Overall framework of DyGPrompt.

Dual prompts

Node prompt

$$\mathbf{x}^{\text{node}}_{t,v} = \mathbf{p}^{\text{node}} \odot \mathbf{x}_{t,v}$$

Time prompt

$$\mathbf{f}^{\text{time}}_{t} = \mathbf{p}^{\text{time}} \odot \mathbf{f}_{t}$$

Dual condition prompts

Time conditioned node prompts

$$\tilde{\mathbf{p}}^{\text{node}}_{t} = \texttt{TCN}(\mathbf{f}^{\text{time}}_{t}; \kappa)$$

$$\tilde{\mathbf{x}}^{\text{node}}_{t,v} = \tilde{\mathbf{p}}^{\text{node}}_{t} \odot \mathbf{x}^{\text{node}}_{t,v}$$

Node conditioned time prompts

$$\tilde{\mathbf{p}}^{\text{time}}_{t,v} = \texttt{NCN}(\mathbf{x}^{\text{node}}_{t,v}; \phi)$$

$$\tilde{\mathbf{f}}^{\text{time}}_{t,v} = \tilde{\mathbf{p}}^{\text{time}}_{t,v} \odot \mathbf{f}^{\text{time}}_{t}$$

Experiment

Table 1: AUC-ROC (%) evaluation of temporal node classification and link prediction.

Methods	Node Classification				Transductive Link Prediction				Inductive Link Prediction			
	Wikipedia	Reddit	MOOC	Genre	Wikipedia	Reddit	MOOC	Genre	Wikipedia	Reddit	MOOC	Genre
GCN-ROLAND	58.86±10.3	48.25±9.57	49.93±6.74	46.33±3.97	49.61±3.12	50.01±2.53	49.82±1.44	49.15±3.74	49.60±2.37	49.90±1.64	49.16±2.48	47.25±2.97
GAT-ROLAND	62.81±9.88	47.95±8.42	50.01±6.34	47.26±3.49	52.34±1.82	50.04±1.98	55.74±3.71	47.69±2.81	52.29±1.97	49.85±2.35	54.01±2.16	49.38±2.72
TGAT	67.00±5.35	53.64±5.50	59.27±4.43	51.26±2.31	55.78±2.03	62.43±1.86	51.49±1.30	69.11±3.89	48.21±1.55	57.30±0.70	51.42±4.27	48.38±4.72
TGN	50.61±13.6	49.54±6.23	50.33±4.47	50.72±2.31	72.48±0.19	67.37±0.07	54.60±0.80	86.46±2.84	74.38±0.29	69.81±0.08	54.62±0.72	87.17±2.68
TREND	69.92±9.27	64.85±4.71	66.79±5.44	50.34±1.62	63.24±0.71	80.42±0.45	58.70±0.78	52.78±1.14	50.15±0.90	65.13±0.54	57.52±1.01	45.31±0.43
GraphMixer	65.43±4.21	60.21±5.36	63.72±4.98	50.15±1.49	59.73±0.35	61.88±0.11	52.42±1.38	60.83±3.25	51.34±0.84	57.64±0.31	51.16±2.59	56.32±3.08
DDGCL	65.15±4.54	55.21±6.19	62.34±5.13	50.91±2.08	54.96±1.46	61.68±0.81	55.62±0.32	68.49±5.31	47.98±1.11	55.90±1.13	55.18±2.73	42.70±3.26
CPDG	43.56±6.41	65.92±6.25	50.32±5.06	49.89±1.34	52.86±0.64	59.72±2.53	53.82±1.50	49.71±2.64	47.37±2.23	56.40±1.17	53.58±2.10	40.01±3.59
GraphPrompt	73.78±5.62	60.89±6.37	64.60±5.76	51.28±2.43	55.67±0.26	67.46±0.31	51.07±0.75	86.78±3.14	48.46±0.28	59.18±0.49	50.27±0.58	87.45±2.5
ProG	60.86±7.43	68.60±5.64	63.18±4.79	51.46±2.38	92.28±0.21	93.32±0.06	58.73±1.58	86.24±2.87	89.75±0.28	90.69±0.08	56.42±1.95	85.43±3.1
TGAT-TIGPrompt	69.21±8.88	67.70±9.64	73.90±6.68	51.38±2.72	59.54±1.41	78.45±1.44	51.69±1.24	69.71±4.16	49.52±0.85	65.66±2.68	51.58±4.02	48.34±3.2
TGN-TIGPrompt	44.80±5.45	63.75±5.60	55.42±3.60	50.84±2.75	82.04±2.03	83.26±2.38	65.00±4.73	86.25±2.43	81.75±1.97	79.51±2.58	64.98±4.61	86.19±3.0
TGAT-DyGPrompt	**82.09**±6.43	73.50±6.47	**77.78**±5.08	**52.03**±2.24	69.88±0.18	90.76±0.09	53.92±0.97	72.04±4.71	52.58±0.23	75.20±0.17	53.29±0.87	50.82±3.6
TGN-DyGPrompt	74.47±3.44	**74.00**±3.10	69.06±3.89	51.97±2.16	**94.33**±0.12	**96.82**±0.06	**70.17**±0.75	**87.02**±1.63	**92.22**±0.19	**95.69**±0.08	**69.77**±0.66	**87.63**±1.9

Results are reported in percent. The best method is bolded and the runner-up is underlined.

Table 2: Ablation study reporting AUC-ROC (%), with TGAT as the backbone.

Methods	Node prompt	Time prompt	NCN	TCN	Node classification			Transductive Link Prediction			Inductive Link Prediction		
					Wikipedia	Reddit	MOOC	Wikipedia	Reddit	MOOC	Wikipedia	Reddit	MOOC
Variant 1	×	×	×	×	67.00	53.64	59.27	55.78	62.43	51.49	48.21	57.30	51.42
Variant 2	✓	×	×	×	72.59	61.82	63.50	68.12	88.59	51.24	51.89	74.84	51.37
Variant 3	×	✓	×	×	73.22	62.51	62.59	66.51	87.06	51.26	50.28	69.71	50.50
Variant 4	✓	✓	×	×	72.25	63.11	62.87	68.36	90.31	52.17	52.56	75.13	51.33
Variant 5	✓	×	✓	×	81.40	73.12	77.15	69.56	90.10	52.16	52.57	75.50	53.31
Variant 6	×	✓	×	✓	80.34	72.59	76.16	66.62	87.34	51.22	49.05	73.16	52.34
DyGPrompt	✓	✓	✓	✓	82.09	73.50	77.78	69.88	90.76	53.92	52.58	75.20	53.29

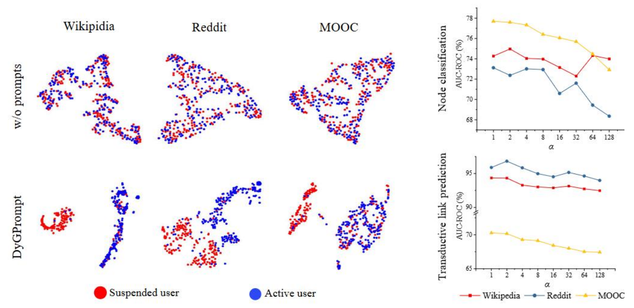


Figure 3: Visualization of output embedding space of nodes.

Figure 4: Sensitivity of α.

Table 3: AUC-ROC (%) evaluation of DyGPrompt with different DGNN backbones.

Pre-training Backbone	Downstream Adaptation	Node classification			Transductive link prediction			Inductive link prediction		
		Wikipedia	Reddit	MOOC	Wikipedia	Reddit	MOOC	Wikipedia	Reddit	MOOC
DyRep	-	50.61	49.54	50.33	58.45	58.02	50.29	56.87	57.25	50.34
	DyGPrompt	**53.52**	**50.98**	**51.62**	**91.64**	**93.84**	**72.04**	**90.28**	**93.08**	**72.27**
JODIE	-	51.37	49.80	50.53	62.40	**59.81**	50.71	59.59	**61.28**	50.57
	DyGPrompt	**62.84**	**60.93**	**67.84**	**63.56**	58.89	**52.06**	**62.80**	58.17	**52.33**
TGAT	-	67.00	53.64	59.27	55.78	62.43	51.49	48.21	57.30	51.42
	DyGPrompt	**82.09**	**73.50**	**77.78**	**69.88**	**90.76**	**53.92**	**52.58**	**75.20**	**53.29**
TGN	-	50.61	49.54	50.33	72.48	67.37	54.60	74.38	69.81	54.62
	DyGPrompt	**74.47**	**74.00**	**69.06**	**94.33**	**96.82**	**70.17**	**92.22**	**95.69**	**69.77**
TREND	-	69.92	64.85	66.79	63.24	**80.42**	58.70	50.15	**65.13**	57.52
	DyGPrompt	**70.15**	**65.24**	**67.58**	**64.35**	79.62	**59.45**	**51.26**	64.88	**59.13**
GraphMixer	-	65.43	60.21	63.72	59.73	61.88	52.42	51.34	**57.64**	51.16
	DyGPrompt	**66.39**	**61.42**	**64.18**	**60.25**	**62.31**	**52.94**	**52.19**	57.43	**52.55**

"-" refers to fine-tuning or continually training the backbone on downstream task data without our prompt design.