GPL4SRec: Graph Multi-Level Aware Prompt Learning for Streaming Recommendation

Hao Cang[1], **Huanhuan Yuan**[1], **Jiaqing Fan**[1], **Lei Zhao**[1], **Guanfeng Liu**[2] and **Pengpeng Zhao**[1*]
[1]School of Computer Science and Technology, Soochow University, China
[2]Department of Computing, Macquarie University, Australia
hcangada@stu.suda.edu.cn, hhyuan@stu.suda.edu.cn, jqfan@suda.edu.cn, zhaol@suda.edu.cn, guanfeng.liu@mq.edu.au, ppzhao@suda.edu.cn

Abstract

Streaming Recommendation (SRec) aims to capture evolving user preferences in the streaming scenarios. Recently, Graph Prompt Learning (GPL) methods have demonstrated their effectiveness and adaptability within SRec. However, existing graph prompt solutions rarely consider the evolution of multi-hop cascading relationships between users and items, which are crucial for modeling the shifts in user preferences. To address this problem, we propose a novel **G**raph Multi-Level Aware **P**rompt **L**earning for **S**treaming **Rec**ommendation, named **GPL4SRec**. Specifically, a graph encoder is first pre-trained on extensive historical data to capture user long-term preferences. Then, we design three types of prompts, namely node-aware, structure-aware, and layer-aware prompts, which are used to guide the pre-trained encoder to better capture user short-term preferences. This is accomplished by accounting for both the incremental changes in users and items, as well as the cascading evolution in multi-hop relationships. Furthermore, we provide a theoretical analysis showing that our prompt templates are critical to achieving superior performance. Finally, experimental results also prove that our model significantly outperforms the state-of-the-art approaches in SRec.

1 Introduction

Recommender systems (RS) have become a fundamental component of online platforms like Amazon, Netflix, and YouTube, driving personalized user experiences and content discovery. However, traditional recommendation models, which are trained on offline static datasets, often suffer from performance degradation when deployed in streaming scenarios [He *et al.*, 2020]. To overcome this limitation, the concept of streaming recommendation has been introduced, focusing on the dynamic updating and deployment of recommendation models in response to continuous data streams [He *et al.*, 2023; Zhang *et al.*, 2024; Yang *et al.*, 2024].

Recently, Graph Neural Networks (GNNs) have stood out as a groundbreaking approach in traditional static recommendation scenarios [He *et al.*, 2020; Guo *et al.*, 2023], relying on their strong capability to capture complex collaborative patterns between users and items. Nevertheless, due to the necessity for continuous model updates in streaming scenarios, these conventional GNN-based recommendation models can not be directly applied to handle streaming data. Therefore, some recent studies have concentrated on two primary areas: dynamic graphs and graph fine-tuning. First, dynamic graph methods, such as EvolveGCN [Pareja *et al.*, 2020] and ROLAND [You *et al.*, 2022], typically utilize recurrent neural networks on the graph to capture and track ever-evolving user preferences. However, these methods struggle to capture rapidly changing user preferences, exhibiting *over-stability* [Ostapenko *et al.*, 2021] in incremental learning where model overly reliant on historical knowledge and fail to adapt to sudden and dramatic shifts. Second, graph fine-tuning approaches, such as DEGC [He *et al.*, 2023] and G-Tuning [Sun *et al.*, 2024], mainly focus on more efficiently updating the parameters of pre-trained graph models as new interaction data arrives, enabling faster and more precise adaptation to the evolving user preferences from new data.

However, the above graph fine-tuning methods may face the problem of *Catastrophic Forgetting* [Chang *et al.*, 2017a] in streaming environments. Recently, a novel method known as graph prompt learning has emerged as an effective and adaptive technique for mitigating the issue of catastrophic forgetting. These methods, such as GPF [Fang *et al.*, 2023] and GraphPrompt [Liu *et al.*, 2023], not only aid in preventing catastrophic forgetting by maintaining the integrity of the model fundamental knowledge but also ensures effective learning and adaptation to new and diverse data patterns through node prompts and structural prompts. Node prompts mainly focus on continual evolutions in individual users and items attributes, while structural prompts primarily emphasize the incremental changes of one-hop user-item relationships. *However*, the design of existing graph prompt templates [Fang *et al.*, 2023; Yang *et al.*, 2024; Liu *et al.*, 2023; Zhang *et al.*, 2024] focusing solely on the incremental changes of nodes and edges is suboptimal. In fact, these changes within the graph are not merely incremental variations in nodes and edges but also cascaded, profoundly influencing the entire graph. This cascading nature of these changes implies that incremental updates simultaneously affect multi-hop cascading relationships within the

* Corresponding author.

Graph Prompt Learning	Node	Structure	Layer
GPF/GPF-plus [Fang *et al.*, 2023]	✓	×	×
GraphPrompt [Liu *et al.*, 2023]	✓	✓	×
GPT4Rec [Zhang *et al.*, 2024]	✓	✓	×
GraphPro [Yang *et al.*, 2024]	✓	✓	×
GPL4SRec (ours)	✓	✓	✓

Table 1: A general comparison of our proposed GPL4SRec with existing graph prompt learning in terms of node, structure and layer.

graph. The neglect of such multi-hop cascading relationships hinders the full integration of the model historical knowledge and new data, which is crucial for more accurately capturing shifts in user preferences. Therefore, as shown in Table 1, designing a prompt template that not only captures incremental changes in nodes and edges but also accounts for multi-hop cascading changes across graph layers presents a unique challenge. To address the problems mentioned above, we propose GPL4SRec, a Graph Multi-Level Aware Prompt Learning for Streaming Recommendation. Specifically, we first pre-train a graph encoder on a large volume of historical data to effectively capture users long-term preferences. Then, we design three distinct types of graph prompts: node-aware, structure-aware, and layer-aware prompts, which serve to guide the pre-trained encoder in more effectively capturing users short-term preferences. This is achieved by accounting for the incremental changes in nodes and interactions, as well as the cascading modifications in multi-hop relationships between users and items. Our theoretical analysis demonstrates that the design of these prompt templates plays a pivotal role in achieving superior performance. Finally, experimental results also demonstrate that our model significantly outperforms the state-of-the-art approaches, especially in mitigating the issue of catastrophic forgetting in SRec. In summary, the main contributions of our work can be concluded as follows:

- We propose a novel graph multi-level aware prompt learning framework for streaming recommendation that effectively mitigates the issue of catastrophic forgetting in streaming scenarios.
- We design three types of graph prompt templates: node-aware, structure-aware, and layer-aware prompts, which effectively guide the model to capture both incremental and cascaded changes within the graph.
- We provide a theoretical analysis showing that our prompt templates are essential for achieving superior performance. Moreover, extensive evaluations on three datasets demonstrate the state-of-the-art performance of our model in streaming recommendation.

2 Related Work

There are multiple topics related to our GPL4SRec. In this section, we first review existing researches on streaming recommendation, followed by a discussion on current dynamic graph learning and graph prompt learning techniques.

2.1 Streaming Recommendation

Traditional RS, trained on static datasets, struggle to accurately capture the evolving user preferences when applied to streaming scenarios. Therefore, streaming recommendation [Chang *et al.*, 2017b; Wang *et al.*, 2018] has been introduced to address this challenge by updating and deploying recommendation model in response to continuous data streams. Early researches [Chandramouli *et al.*, 2011; Lommatzsch and Albayrak, 2015] primarily focused on analyzing the popularity, recency, and trends of items , followed by the introduction of matrix decomposition, collaborative filtering, and other methods combined with online clustering technology [Li *et al.*, 2019; Li *et al.*, 2016]. Recently, GNNs have become a research hots pot in the field of recommendations due to their powerful relationship modeling capabilities [Ying *et al.*, 2018; Chen *et al.*, 2020; Chen and Wong, 2020]. However, the challenge lies in adapting GNNs, initially trained on static datasets, to effectively manage the continuous data stream in streaming recommendation scenarios [Wang *et al.*, 2020; Xu *et al.*, 2020b].

2.2 Dynamic Graph Learning

Dynamic graph learning has attracted significant attention in recent years, with researches primarily categorized into snapshot-based and event-based methods, depending on their temporal granularity. Snapshot-based approaches (such as EvolveGCN, DGCN, and ROLAND [Pareja *et al.*, 2020; Li *et al.*, 2020; You *et al.*, 2022]) construct dynamic graphs from scratch and leverage recurrent neural networks to capture temporal changes. While effective in modeling smooth transitions, these methods often struggle with abrupt shifts in user preferences and are prone to noise in user and item representations [Ostapenko *et al.*, 2021], thereby limiting their applicability in streaming environments. In contrast, event-based methods [Trivedi *et al.*, 2019; Ma *et al.*, 2020; Xu *et al.*, 2020a] focus on capturing graph evolution at finer temporal granularity, but challenges remain in balancing computational efficiency and temporal precision.

2.3 Graph Prompt Learning

Graph prompt learning has emerged as a powerful technique due to its adaptability and effectiveness in graph tasks. Recently, several significant advancements in this area have been proposed. GraphPrompt [Liu *et al.*, 2023] introduces a universal prompt template with a learnable readout function, enabling the unification of multiple downstream tasks. GPF [Fang *et al.*, 2023] proposes a universal prompt-based tuning method for pre-trained GNNs, aligning pre-training and downstream tasks through feature-space adjustments, without the need for task-specific prompt designs. GPT4Rec [Zhang *et al.*, 2024] first introduces graph prompt learning for streaming recommendation through designing special prompt templates. GraphPro [Yang *et al.*, 2024] incorporates time-aware and structural prompts for dynamic recommendations, extending the capabilities of pre-trained models to accommodate evolving data. Although these methods have achieved

acceptable results, the suboptimal design of existing graph prompt templates remains a critical challenge. Our research complements the shortcomings of existing prompt templates and designs three types of prompts taking into account both the incremental and cascading changes within graph, which can better guide the seamless integration of historical knowledge and new data for mitigating the problem of catastrophic forgetting in streaming recommendation.

3 Preliminaries

In this section, we first formalize the task definition for streaming recommendation. Then we briefly introduce the definition of graph incremental learning for streaming recommendation used in this paper.

Definition 1. Streaming Recommendation. Real-world RS must process a continuous stream of user-item interaction data, denoted as D. This continuous data stream is partitioned into sequential snapshots [D_1,...,D_{t-1},D_t,...,D_T] with an equal time span. At each time snapshot t, the model needs to optimize its performance on D_t by combining previous knowledge from historical snapshots [D_1,...,D_{t-1}]. The recommendation performance is subsequently evaluated across the entire timeline.

Definition 2. Graph Incremental Learning. Given the data stream defined above, the graph snapshots G are represented as [G_1,...,G_{t-1},G_t,...,G_T] on the data snapshots D, where $G_t = G_{t-1} + \Delta G_t$. $G_t = (A_t, X_t)$ is an attributed graph at time t, where A_t and X_t are the adjacency matrix and node features of G_t. $\Delta G_t = (\Delta A_t, \Delta X_t)$ represents the changes of graph structures and node information at time t. Therefore, the goal of graph incremental learning (GIL) is learn $\Delta G_t(D_t)$ sequentially while transferring previous knowledge to new graph segments effectively. Mathematically, GIL aims to learn the optimal graph structure S_t and paramters W_t at each segment t, formulated as:

$$(\mathrm{S}_t^*, \mathrm{W}_t^*) = \underset{(\mathrm{S}_t, \mathrm{W}_t)}{\arg\min} \mathcal{L}_t\,(\,\mathrm{S}_t, \mathrm{W}_t, \Delta G_t) \tag{1}$$

where $(S_t, W_t) \in (S, W)$. The S and W are corresponding search spaces, respectively. The function $L_t(S_t, W_t, \Delta G_t)$ denotes the loss for the snapshot t, evaluated on ΔG_t.

4 Methodology

The GPL4SRec framework shown as Figure 1 consists of two main stages: graph pre-training learning and graph prompt learning. During graph pre-training, we pre-train a graph encoder on massive historical interaction data, enabling it to effectively capture user long-term preferences. During graph prompt learning stage, we primarily focus on designing node-aware, structure-aware and layer-aware prompts to capture multi-level changes within the graph, further adapting user evolving short-term preferences. Finally, theoretically analyze the critical role of prompt template optimization in achieving improved performance.

4.1 Graph Pre-training

To better capture the evolving nature of user preferences, we decompose them into long-term preferences (LTP) and short-term preferences (STP). LTP generally remains stable over time, influencing multiple snapshots and is often shaped by factors such as gender, occupation, family and education. In contrast, STP exhibits rapid fluctuations and is more relevant within specific time windows, typically driven by contextual factors such as user emotions. In our framework, the model is designed to learn and integrate both LTP and STP within its parameters, adapting to the dynamic shifts in user preferences over time. Therefore, to effectively capture and encode LTP, we pre-train a graph encoder using extensive historical interaction data, ensuring that LTP is accurately reflected in the user embeddings. The learning objective is based on contrastive learning, optimizing the LTP representation:

$$\mathcal{L}_P = -\sum_{u \in V} \ln\left(\frac{\sum_{a \in \mathcal{P}_u} \mathrm{s}(h_u, h_a)}{\sum_{a \in \mathcal{P}_u} \mathrm{s}(h_u, h_a) + \sum_{b \in \mathcal{N}_u} \mathrm{s}(h_u, h_b)}\right) \tag{2}$$

where s($\cdot$, $\cdot$) represents a similarity function such as inner product in our experiment, P_u and N_u are respectively the set of positive instances and negative instances for u. Within our proposed framework, we incorporate the Bayesian Personalized Ranking (BPR) [Rendle *et al.*, 2012] algorithm in conjunction with other graph self-supervised learning strategies [Wu *et al.*, 2021; Huang *et al.*, 2021].

4.2 Graph Prompt Learning

After grasping LTP through graph pre-training representation learning, we mainly focus on capturing STP via graph prompt learning. Specifically, we design node-aware, structure-aware and layer-aware prompt to capture the comprehensive essence of graph patterns in streaming recommendation.

Node Aware Prompt Template Design

Node-aware prompts focus on the attributes or properties of individual nodes within the graph, such as user and item characteristics in recommendation models. By emphasizing the node level, GPL4SRec effectively captures the nuances of node-specific data, providing a deeper understanding of user behaviors and item features. This level of analysis is crucial for tasks that require personalized recommendations or detailed attribute evaluation.

Specifically, the node aware prompts consist of a set of learnable parameters, denoted as $NP = [np_1, ..., np_n] \in \mathbb{R}^{n \times f}$, where n is the number of node prompts and f is the feature dimension. These node prompts serve as targeted cues, guiding the model in interpreting and integrating new information related to users or items. We first employ an attentional mechanism based on softmax function to help automatically determine how these prompts transform each node's representation. We then introduce an additional learnable prompt weight λ_i to further refine the node prompting mechanism and help prevent the model from overfitting to noise and irrelevant information in the new data. The λ_i can be learned by using a simple multi-layer perceptron (MLP) based on the node feature as follow:

$$\overline{x}_i = x_i + \lambda_i \odot \sum\nolimits_{j=1}^{n} \frac{\exp(np_j^T x_i)}{\sum_{r=1}^{n} \exp(np_r^T x_i)} np_j \tag{3}$$

$$\lambda_i = \sigma(f_{\text{mlp}}(x_i, w)) \tag{4}$$

where the w represents the learnable parameters which are shared across all nodes and σ denotes the sigmoid function to

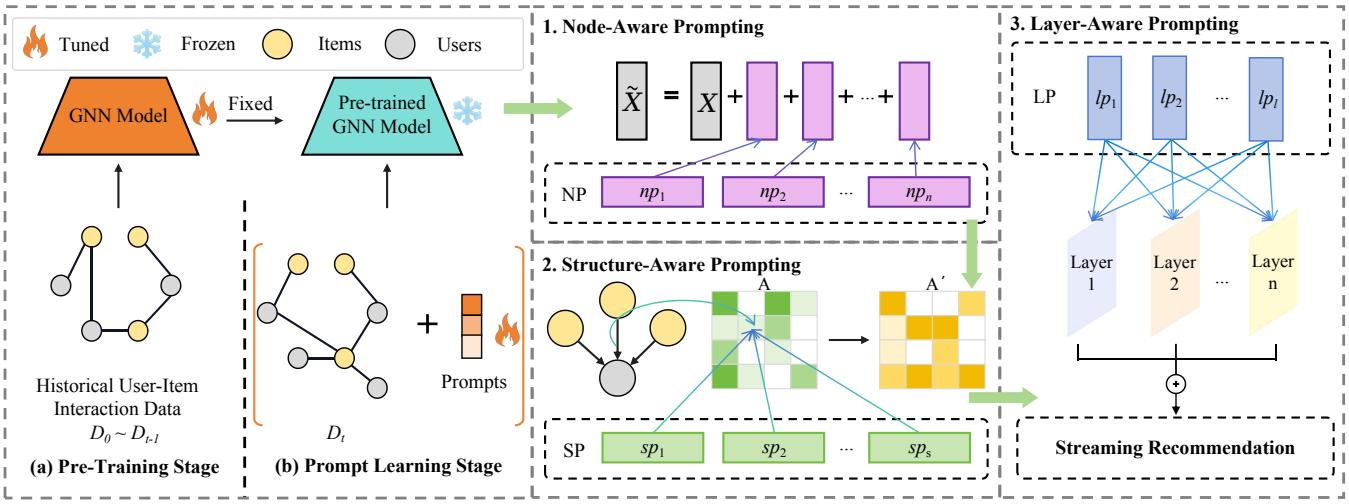

Figure 1: Overview of the GPL4SRec based on "graph pre-training, prompt-tuning" framework. During the prompt-tuning phase, a) Firstly, we design the node-aware prompts to depict the node-level changes. b) Secondly, we propose the structure-aware prompts for adapting to changes in one-hop relationships. c) Finally, we present the layer-aware prompts to capture the changes in multi-hop cascading relationships.

enforce $\lambda_i \in [0, 1]$, denoting the confidence of node x_i to be prompted. Therefore, the prompted node is $\overline{X} = [\overline{x}_i]$, which represents the search space of node. These prompts undergo an attention mechanism that selectively adjusts the weights of data most pertinent to the current node.

Structure Aware Prompt Template Design

In addition to node aware prompts, structure aware prompts are devised to interact with the more extensive connectivity and relationship patterns within the graph. These prompts play a vital role in comprehending and adjusting to alterations in the overall graph topology, like the appearance of novel interaction patterns or the development of existing ones.

The structure aware prompt is designed as follows: a set of learnable parameters, denoted as $SP = [sp_1, ..., sp_s] \in \mathbb{R}^{s \times f}$ for the nodes that adaptively aggregate the structure information via prompting adjacency matrix $A = [a_{ij}]$. To be specific, given a node x_i, the node $\hat{x}_i$ prompted by structure aware prompts can be computed:

$$\hat{x}_i = \overline{x}_i + \sum_{j \in N(x_i)} \overline{a}_{ij} \overline{x}_j \tag{5}$$

$$\overline{a}_{ij} = a_{ij} + \sigma\left(\sum_{j=1}^{s} \frac{\exp(sp_j^T e_{ij})}{\sum_{r=1}^{s} \exp(sp_r^T e_{ij})} sp_j\right) \odot a_{ij} \tag{6}$$

where $e_{ij} = f_{mlp}(x_i \| x_j \| t_{ij})$ denotes the feature of edge (i, j) which is obtained by a MLP based on node x_i, x_j and interaction time t_{ij}. The prompted structure $\overline{A} = [\overline{a}_{ij}]$ represents the search space of structure. Therefore, these prompts can guide models to capture dynamic structural changes for adapting to the evolving nature of the data.

Layer Aware Prompt Template Design

The importance of node and structural information on the graph is widely recognized. Nevertheless, current research has also revealed that the relationships among layers are extremely critical, as different layers focus on distinct information. For example, in recommendation models, different layers respectively pay attention to the relationships between user-user, user-item and item-item. Therefore, designing a layer-aware prompt that can better capture the cascading relationships among layers poses a great challenge.

To address this challenge, we design the layer aware prompts as follows: a set of learnable parameters, denoted as $LP = [lp_1, ..., lp_l] \in \mathbb{R}^{l \times f}$ for the nodes via prompting each graph layer embedding x_i^k. Given a node x_i, the node $\tilde{x}_i$ prompted by layer aware prompts can be computed:

$$\tilde{x}_i = \hat{x}_i + \sum_{k=1}^{K} \beta_i^k \hat{x}_i^k \tag{7}$$

$$\beta_i^k = \sum_{j=1}^{l} \alpha_{ij}^k lp_j \tag{8}$$

$$\alpha_{ij}^k = \frac{\exp\left(lp_j^T \hat{x}_i^k\right)}{\sum_{r=1}^{l} \exp\left(lp_r^T \hat{x}_i^k\right)} \tag{9}$$

where K denotes the total layers of GNNs and x_i^k represents the k^{th} layer representation of x_i. $\overline{L} = [\beta_i^k]$ represents the search space of layers. These prompts will motivate the model to learn the multi-hop relationship between the different layers of node x_i and better extract the relevant node information in the deeper layers of the GNNs.

4.3 Theoretical Analysis

In this section, we conduct theoretical analysis to guarantee the correctness of the proposed graph prompt learning algorithm within the context of streaming recommendation.

THEOREM 1: *GPL4SRec's performance is determined by our design of prompt templates, and it satisfies at least the ability of fine-tuning globally using full data.*

PROOF: Given a pre-trained graph encoder f and a streaming task including graph and dataset over time $D = \{(G_1, Y_1), .., (G_t, Y_t), (G_{t+1}, Y_{t+1}), ..., (G_T, Y_T)\}$, we update the pre-trained model f_{t+1} to maximize the likelihood

of predicting the correct labels Y_{t+1} at the time $t+1$:

$$\arg\max_{f_{t+1}} P_{f_{t+1}}(Y_{t+1}|G_{t+1}) \tag{10}$$

where we use f_t to initialize model parameters f_{t+1}. The optimization has the upper bound as [Zhang *et al.*, 2023]:

$$\arg\max_{f_{t+1}} P_{f_{t+1}}(Y_{t+1}|\Delta G_t) + \arg\max_{f_t} P_{f_t}(Y_t|G_t) \tag{11}$$

where ΔG_t denotes the graph gap and the optimization process after using our prompt template is:

$$\arg\max_{\phi} P_{f_{t+1}}(Y_{t+1}|g_\phi(\Delta G_t)) + \arg\max_{f_t} P_{f_t}(Y_t|G_t) \tag{12}$$

where $g_\phi(\Delta G_t)$ is the our proposed graph prompt templates including node, structure and layer levels. Hence, the final object function for the prompt is:

$$\begin{aligned}&\arg\max_{\overline{A}\in\mathbb{A},\overline{X}\in\mathbb{X},\overline{L}\in\mathbb{L}} P_{f_{t+1}}(Y_{t+1}|\text{prompt}(\Delta A_t, \Delta X_t, \Delta L_t)) + \\ &\arg\max_{f_t} P_{f_t}(Y_t|G_t)\end{aligned} \tag{13}$$

where $(\mathbb{A}, \mathbb{X}, \mathbb{L})$ is the candidate search spaces of prompted $(\overline{A}, \overline{X}, \overline{L}) = \text{prompt}(\Delta A_t, \Delta X_t, \Delta L_t)$. $(\Delta A_t, \Delta X_t, \Delta L_t)$ is the incremental and cascading changes in G_t. Equation 13 shows that fixing f_t and then prompting the graph gap ΔG_t is equivalent to directly optimizing Equation 11, which is the upper bound of Equation 10. Therefore, Equation 13 can show that our proposed graph prompt learning method satisfies at least Equation 10 fining-tuning with full data and its performance is determined by our prompt templates. □

5 Experiments

In this section, we conduct experiments with the aim of answering the following questions: **Q1:** How do our proposed GPL4SRec perform compared with other baselines? **Q2:** What is the influence of key components of GPL4SRec? **Q3:** How is the robustness of GPL4SRec? **Q4:** Whether is GPL4SRec sensitive to the hyper-parameters? **Q5:** How efficient is the training of GPL4SRec in streaming scenarios?

5.1 Experimental Settings

Datasets. We use three public datasets that cover diverse real-world scenarios in streaming recommendation. The Taobao dataset records the implicit feedback from taobao.com, which is a Chinese e-commerce platform, during a period of 10 days. The Koubei dataset, which is provided for the IJCAI'16 contest, documents 9 weeks' worth of user interactions with local stores on Koubei within Alipay. The Amazon dataset is composed of a collection of product reviews from Amazon that spans 13 weeks. Detailed information about these datasets can be found in Table 2.

Evaluation Protocols. In our evaluation, we simulate real-world dynamics using graph snapshots taken at different intervals (weekly/daily). We use a two-step sliding window to learn from current data and predict future changes. Following the Pre-train and Fine-tune paradigm, we pre-train on most of the dataset, fine-tune, and evaluate on later snapshots (see Table 2). For consistency, the same method is applied to all baselines. Dynamic GNNs start fine-tuning with pre-training weights. Results are averaged on future snapshots, and standard metrics like Recall@k and NDCG@k (k = 10 and 20) are applied, in line with prior work [Yang *et al.*, 2024].

Statistics	Amazon	Koubei	Taobao
# No. of users	131K	119K	117K
# No. of items	107K	101K	86K
# No. of interactions	876K	3986K	8795K
Temporal Segmentation			
# Pre-training Span	4 weeks	4 weeks	5 days
# Tuning-Predicting Span	9 weeks	5 weeks	5 days
# Snapshot Granularity	weekly	weekly	daily

Table 2: Statistics and temporal segmentation of experiment dataset.

Baseline Methods. We include the recent dynamic graph learning methods, graph prompt learning approaches, and traditional graph fine-tuning as our baselines:

- **Dynamic Graph Neural Networks.** These graph neural networks are tailored to dynamic scenarios, updating embedding with time sensitivity to reflect graph changes. We benchmark our approach against notable models: **EvolveGCN-O and EvolveGCN-H** [Pareja *et al.*, 2020] and **ROLAND** [You *et al.*, 2022].
- **Graph Prompt Learning Methods.** This line focuses on leveraging prompts to get task-specific knowledge related to downstream tasks, thereby becoming a unified approach for downstream tasks. We compare our approach with notable models: **GPF/GPF-Plus** [Fang *et al.*, 2023], **GraphPrompt** [Liu *et al.*, 2023], **GPT4Rec** [Zhang *et al.*, 2024], and **GraphPro** [Yang *et al.*, 2024].

Implementation Details. We run all methods in PyTorch [Paszke *et al.*, 2017] with Adam [Diederik, 2014] optimizer on an NVIDIA GeForce 4070Ti GPU. In our experiment, the batch size b and the demension of embeddings d are set 2048 and 64. The layers k of GNNs are set 3. We train all models 300 epochs at every snapshot. We apply grid search to find the optimal hyper-parameters for each model. The ranges of hyper-parameters are [16, 32, 64, 128] for the size N of node-aware prompts NP, the size S of structure-aware prompts SP and the size of layer-aware prompts LP. GPL4SRec is trained with a learning rate of 0.001.

5.2 Overall Perference (Q1)

In this section, we report the experimental results of different methods in the Table 3, where R and N are abbreviations for Recall and NDCG, respectively. It can be clearly observed that our proposed GPL4SRec surpasses both graph prompt learning methods and dynamic graph learning methods, highlighting the effectiveness of our graph pre-training strategy and graph prompt learning methods.

First, this superior performance of GPL4SRec can be attributed to two key factors: 1) Our graph pre-training representation learning that adeptly captures and encodes constant LTP of users during pre-training stage, 2) Our graph multi-level aware prompt templates design, which includes three types of prompts: node-aware, structure-aware, and layer-aware prompts. These well-designed prompts ensure seam-

Dataset	Metric	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	**(j)**	**(k)**
		Finetune	EvolveGCN-O	EvolveGCN-H	ROLAND	GraphPrompt	GPF	GPF-Plus	GPT4Rec	GraphPro	**GPL4SRec**	**Improv.**
Amazon	R@10	0.0114	0.0115	0.0098	0.0097	0.0094	0.0120	0.0115	0.0125	0.0129	**0.0138**	**6.98%**
	R@20	0.0172	0.0157	0.0138	0.0150	0.0154	0.0174	0.0172	0.0185	0.0191	**0.0202**	**5.76%**
	N@10	0.0069	0.0070	0.0057	0.0055	0.0056	0.0072	0.0069	0.0070	0.0076	**0.0083**	**9.21%**
	N@20	0.0087	0.0084	0.0066	0.0069	0.0075	0.0088	0.0087	0.0089	0.0094	**0.0099**	**5.32%**
Koubei	R@10	0.0216	0.0206	0.0201	0.0195	0.0212	0.0214	0.0214	0.0230	0.0231	**0.0239**	**3.46%**
	R@20	0.0344	0.0334	0.0315	0.0301	0.0342	0.0348	0.0345	0.0358	0.0362	**0.0368**	**1.66%**
	N@10	0.0199	0.0190	0.0187	0.0188	0.0196	0.0199	0.0198	0.0214	0.0216	**0.0221**	**2.31%**
	N@20	0.0249	0.0242	0.0231	0.0223	0.0249	0.0251	0.0250	0.0261	0.0265	**0.0269**	**1.51%**
Taobao	R@10	0.0131	0.0138	0.0132	0.0135	0.0123	0.0130	0.0129	0.0138	0.0142	**0.0150**	**5.63%**
	R@20	0.0223	0.0236	0.0224	0.0226	0.0199	0.0223	0.0222	0.0245	0.0251	**0.0262**	**4.38%**
	N@10	0.0199	0.0211	0.0203	0.0210	0.0188	0.0200	0.0196	0.0212	0.0216	**0.0225**	**4.17%**
	N@20	0.0218	0.0232	0.0221	0.0226	0.0195	0.0220	0.0216	0.0238	0.0245	**0.0259**	**5.71%**

Table 3: The average performance with LightGCN as our base model. The numbers in bold indicate statistically significant improvement ($p < 0.01$) by the pairwise t-test comparisons over the other baselines.

less knowledge transfer from the pre-trained model across temporal snapshots. Such adaptive learning helps model to better capture incremental and cascaded changes such as STP of users in the graph, which is very important to understand user behavior in streaming recommendation.

Second, there are more findings in these comparative experiments. The different performance of baseline methods shows the complexity of streaming recommendations. Graph prompt learning methods broadly outperform dynamic graphs and fine-tuning methods, indicating the advantage of graph prompt learning in streaming recommendation, especially in mitigating catastrophic forgetting.

Finally, different graph prompt learning methods show certain performance differences, and our proposed GPL4SRec exhibits better performance than other prompt learning methods. We argue that the key point of differences is the design of graph prompt templates. Specifically, GPL4SRec is carefully designed with three types of prompts, which help to comprehensively capture the incremental and cascading changes within the graph and more effectively integrate the historical knowledge and new data, thus generating new and complete knowledge for streaming recommendation.

5.3 Ablation Study (Q2)

In this section, we focus on GPL4SRec and test the efficacy of its various designs in regard to the node-aware, structure-aware, and layer-aware prompts. The results are shown in Table 4. We have the following observations:

- **w/o NP.** We studied the effect of node-aware prompts in adaptation to node changes. Comparing prompted and unprompted models, the former responded better to user and item changes, showing node-aware prompts aid context adaptation for improved personalization.
- **w/o SP.** We measured the impact of structure prompts on adapting to graph incremental changes. Our results demonstrated that these prompts accurately represent structural patterns, highlighting their significance in capturing one-hop graph relationships between users and items.

Variants		w/o NP	w/o SP	w/o LP	GPL4SRec
Amazon	R@20	0.0198	0.0195	0.0172	**0.0202**
	N@20	0.0096	0.0093	0.0085	**0.0099**
Koubei	R@20	0.0362	0.0353	0.0352	**0.0368**
	N@20	0.0265	0.0258	0.0256	**0.0269**
Taobao	R@20	0.0259	0.0255	0.0250	**0.0262**
	N@20	0.0250	0.0249	0.0245	**0.0259**

Table 4: The ablation study of GPL4SRec on three different datasets.

- **w/o LP.** We focused on exploring the effect of layer-aware prompts. Our ablation studies have demonstrated that such prompts are capable of capturing cascading relationships from diverse perspectives within the graph, thus enabling a more thorough comprehension of graph data.

5.4 Robustness Analysis (Q3)

Within the framework of GPL4SRec, graph pre-training is of critical significance for capturing user LTP and for enhancing the adaptability of the proposed graph prompt learning method. To conduct a comprehensive assessment of its robustness and adaptability, a systematic exploration of a variety of pre-training strategies was implemented, with MixGCF [Huang *et al.*, 2021] and SGL [Wu *et al.*, 2021] incorporated. This exploration encompassed two core objectives: on the one hand, to verify the performance of GPL4SRec under diverse experimental settings; on the other hand, to guarantee the consistency of objectives between pre-training and prompt tuning, which is of utmost importance for bolstering the overall effectiveness of the model. As shown in the experimental data presented in Table 5 and Table 6, GPL4SRec exhibits remarkable robustness across a wide range of pre-training strategies. It not only attains substantial enhancements in predictive accuracy but also demonstrates outstanding adaptability in complex graph scenarios. Notably, the integration of MixGCF and SGL contributes to strengthening the generalization capabilities of model while simultaneously maintaining its stability, thereby further highlighting the remarkable superiority of GPL4SRec in efficiently handling

Model	Amazon		Koubei	
	R@20	N@20	R@20	N@20
Finetune	0.0184	0.0094	0.0378	0.0278
EvolveGCN-O	0.0171	0.0085	0.0375	0.0276
EvolveGCN-H	0.0129	0.0061	0.0354	0.0262
ROLAND	0.0152	0.0072	0.0349	0.0260
GraphPrompt	0.0180	0.0089	0.0377	0.0276
GPF	0.0182	0.0092	0.0380	0.0278
GPF-Plus	0.0184	0.0094	0.0376	0.0276
GPT4Rec	0.0209	0.0106	0.0380	0.0279
GraphPro	0.0216	0.0109	0.0393	0.0291
GPL4Rec (Ours)	**0.0226**	**0.0112**	**0.0398**	**0.0296**

Table 5: The average performance with MixGCF as our base model. The numbers in bold indicate statistical improvement ($p < 0.01$) by the pairwise t-test comparisons over the other baselines.

Model	Amazon		Koubei	
	R@20	N@20	R@20	N@20
Finetune	0.0190	0.0097	0.0358	0.0265
EvolveGCN-O	0.0173	0.0090	0.0365	0.0268
EvolveGCN-H	0.0137	0.0066	0.0358	0.0263
ROLAND	0.0161	0.0078	0.0340	0.0251
GraphPrompt	0.0161	0.0079	0.0355	0.0261
GPF	0.0187	0.0096	0.0363	0.0266
GPF-Plus	0.0191	0.0097	0.0356	0.0264
GPT4Rec	0.0210	0.0107	0.0358	0.0266
GraphPro	0.0221	0.0114	0.0371	0.0277
GPL4Rec (Ours)	**0.0231**	**0.0119**	**0.0378**	**0.0280**

Table 6: The average performance with SGL as our base model. The numbers in bold indicate statistical improvement ($p < 0.01$) by the pairwise t-test comparisons over the other baselines.

complex streaming recommendation tasks based on GNNs.

5.5 Hyper-parameter Sensitivity (Q4)

We conducted a detailed hyper-parameter study on the model with a particular focus on the prompt size. Specifically, we explored the impact of varying the node-aware , structure-aware, and layer-aware prompt sizes on the Amazon dataset. As shown in Figure 2, in the initial stage, adding the prompt size was found to enhance the performance of model. More prompts could represent complex user-item interactions in a more comprehensive manner, facilitating the model capture of user preferences and item characteristics. However, when the prompt size surpassed a specific threshold, further increments led to diminishing returns. The additional size not only scarcely augmented the explanatory capacity of model but also exerted negligible influence on performance improvement and even caused unnecessary computational overhead. This underlines the significance of precisely ascertaining the optimal prompt size to strike a balance between performance and computational efficiency. It further implies that there exists an optimal range for the prompt size, and exceeding this range would prove detrimental. Future research could focus on delineating this range and its mechanisms.

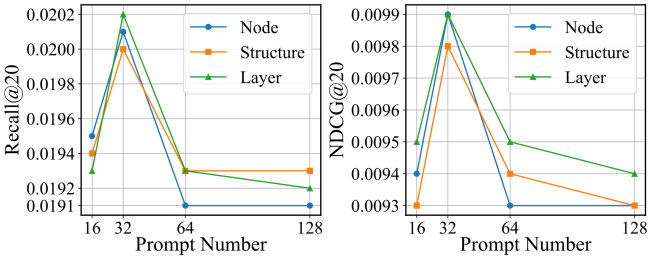

Figure 2: The influence of graph prompt number on Amazon dataset.

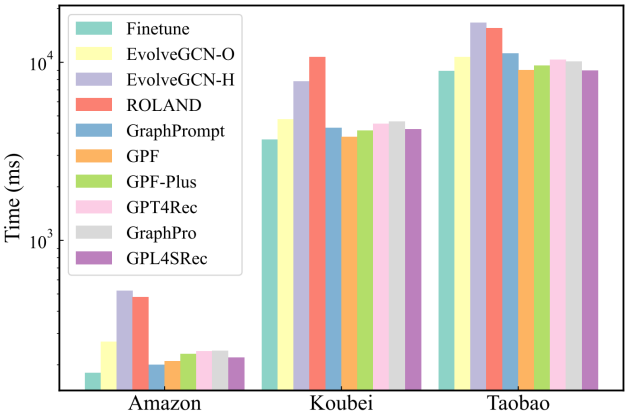

Figure 3: The average training time per epoch on different models.

5.6 Efficiency (Q5)

As shown in Figure 3, we presented the average training time of different models on three datasets per epoch. GPL4SRec rivals fine-tuning and prompt-learning methods in terms of efficiency and surpasses several advanced models in performance. The high efficiency stems from using lightweight graph prompts, which integrate new data seamlessly with low cost, cutting training time and enhancing overall efficiency. Moreover, compared with existing dynamic graph methods, GPL4SRec skips complex graph structure adjustments, averting extra computational costs from frequent reconstruction. Thus, it keeps an efficient training rhythm with large-scale data, strengthening its edge in efficiency. This makes GPL4SRec a highly promising option for scenarios where rapid model training and deployment are crucial.

6 Conclusion

This study proposes GPL4SRec, a graph multi-level aware prompt learning framework for streaming recommendation task. First, GPL4SRec utilized a graph encoder pre-training on extensive historical data to capture the LTP of users. Then, by introducing node-aware, structure-aware, and layer-aware prompts, GPL4SRec could focus on the incremental and multi-hop cascading changes between users and items for accurately capturing the STP of users, thereby mitigating the issue of catastrophic forgetting. Further, our theoretical analysis verified the importance of our prompt template design for GPL4SRec's superiority. Finally, our experiments on three public datasets and three base models confirmed its effectiveness, outperforming existing approaches.

Acknowledgments

This research was partially supported by the NSFC (62376180, 62176175), the National Key Research and Development Program of China (2023YFF0725002), Suzhou Science and Technology Development Program (SYG202328), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[Chandramouli *et al.*, 2011] Badrish Chandramouli, Justin J Levandoski, Ahmed Eldawy, and Mohamed F Mokbel. Streamrec: a real-time recommender system. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1243–1246, 2011.

[Chang *et al.*, 2017a] Shiyu Chang, Yang Zhang, Jiliang Tang, Dawei Yin, Yi Chang, Mark A Hasegawa-Johnson, and Thomas S Huang. Streaming recommender systems. In *Proceedings of the 26th international conference on world wide web*, pages 381–389, 2017.

[Chang *et al.*, 2017b] Shiyu Chang, Yang Zhang, Jiliang Tang, Dawei Yin, Yi Chang, Mark A Hasegawa-Johnson, and Thomas S Huang. Streaming recommender systems. In *Proceedings of the 26th international conference on world wide web*, pages 381–389, 2017.

[Chen and Wong, 2020] Tianwen Chen and Raymond Chi-Wing Wong. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1172–1180, 2020.

[Chen *et al.*, 2020] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 27–34, 2020.

[Diederik, 2014] P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014.

[Fang *et al.*, 2023] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2023.

[Guo *et al.*, 2023] Jiayan Guo, Lun Du, Xu Chen, Xiaojun Ma, Qiang Fu, Shi Han, Dongmei Zhang, and Yan Zhang. On manipulating signals of user-item graph: A jacobi polynomial-based graph collaborative filtering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 602–613, 2023.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[He *et al.*, 2023] Bowei He, Xu He, Yingxue Zhang, Ruiming Tang, and Chen Ma. Dynamically expandable graph convolution for streaming recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1457–1467, 2023.

[Huang *et al.*, 2021] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 665–674, 2021.

[Li *et al.*, 2016] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

[Li *et al.*, 2019] Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[Li *et al.*, 2020] Xiaohan Li, Mengqi Zhang, Shu Wu, Zheng Liu, Liang Wang, and S Yu Philip. Dynamic graph collaborative filtering. In *2020 IEEE international conference on data mining (ICDM)*, pages 322–331. IEEE, 2020.

[Liu *et al.*, 2023] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 417–428, 2023.

[Lommatzsch and Albayrak, 2015] Andreas Lommatzsch and Sahin Albayrak. Real-time recommendations for user-item streams. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1039–1046, 2015.

[Ma *et al.*, 2020] Yao Ma, Ziyi Guo, Zhaocun Ren, Jiliang Tang, and Dawei Yin. Streaming graph neural networks. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 719–728, 2020.

[Ostapenko *et al.*, 2021] Oleksiy Ostapenko, Pau Rodriguez, Massimo Caccia, and Laurent Charlin. Continual learning via local module composition. *Advances in Neural Information Processing Systems*, 34:30298–30312, 2021.

[Pareja *et al.*, 2020] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5363–5370, 2020.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[Sun *et al.*, 2024] Yifei Sun, Qi Zhu, Yang Yang, Chunping Wang, Tianyu Fan, Jiajun Zhu, and Lei Chen. Fine-tuning graph neural networks by preserving graph generative patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9053–9061, 2024.

[Trivedi *et al.*, 2019] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

[Wang *et al.*, 2018] Weiqing Wang, Hongzhi Yin, Zi Huang, Qinyong Wang, Xingzhong Du, and Quoc Viet Hung Nguyen. Streaming ranking based recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 525–534, 2018.

[Wang *et al.*, 2020] Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1515–1524, 2020.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[Xu *et al.*, 2020a] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

[Xu *et al.*, 2020b] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2861–2868, 2020.

[Yang *et al.*, 2024] Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. Graphpro: Graph pre-training and prompt learning for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3690–3699, 2024.

[Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[You *et al.*, 2022] Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2358–2366, 2022.

[Zhang *et al.*, 2023] Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. Continual learning on dynamic graphs via parameter isolation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–611, 2023.

[Zhang *et al.*, 2024] Peiyan Zhang, Yuchen Yan, Xi Zhang, Liying Kang, Chaozhuo Li, Feiran Huang, Senzhang Wang, and Sunghun Kim. Gpt4rec: Graph prompt tuning for streaming recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1774–1784, 2024.