GPT4Rec: Graph Prompt Tuning for Streaming Recommendation

Peiyan Zhang*
Hong Kong University of
Science and Technology
Hong Kong
pzhangao@cse.ust.hk

Yuchen Yan*
School of Intelligence Science and
Technology, Peking University
Beijing, China
2001213110@stu.pku.edu.cn

Xi Zhang
Interdisciplinary Institute for Medical
Engineering, Fuzhou University
Fuzhou, China
zxwinner@gmail.com

Liying Kang
Hong Kong Polytechnic University
Hong Kong
lykangc12@gmail.com

Chaozhuo Li†
Microsoft Research Asia
Beijing, China
lichaozhuo1991@gmail.com

Feiran Huang
Jinan University
Guangzhou, China
huangfr@jnu.edu.cn

Senzhang Wang
Central South University
Changsha, China
szwang@csu.edu.cn

Sunghun Kim
Hong Kong University of
Science and Technology
Hong Kong
hunkim@cse.ust.hk

ABSTRACT

In the realm of personalized recommender systems, the challenge of adapting to evolving user preferences and the continuous influx of new users and items is paramount. Conventional models, typically reliant on a static training-test approach, struggle to keep pace with these dynamic demands. Streaming recommendation, particularly through continual graph learning, has emerged as a novel solution, attracting significant attention in academia and industry. However, existing methods in this area either rely on historical data replay, which is increasingly impractical due to stringent data privacy regulations; or are inability to effectively address the over-stability issue; or depend on model-isolation and expansion strategies, which necessitate extensive model expansion and are hampered by time-consuming updates due to large parameter sets. To tackle these difficulties, we present GPT4Rec, a **G**raph **P**rompt **T**uning method for streaming **Rec**ommendation. Given the evolving user-item interaction graph, GPT4Rec first disentangles the graph patterns into multiple views. After isolating specific interaction patterns and relationships in different views, GPT4Rec utilizes lightweight graph prompts to efficiently guide the model across varying interaction patterns within the user-item graph. Firstly, node-level prompts are employed to instruct the model to adapt to changes in the attributes or properties of individual nodes within the graph. Secondly, structure-level prompts guide the model in adapting to broader patterns of connectivity and relationships within the graph.

*Both authors contributed equally to this research.
†Chaozhuo Li is the corresponding author

Finally, view-level prompts are innovatively designed to facilitate the aggregation of information from multiple disentangled views. These prompt designs allow GPT4Rec to synthesize a comprehensive understanding of the graph, ensuring that all vital aspects of the user-item interactions are considered and effectively integrated. Experiments on four diverse real-world datasets demonstrate the effectiveness and efficiency of our proposal.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Data mining**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Streaming Recommendation, Continual Learning, Graph Prompt Tuning

1 INTRODUCTION

Recommender Systems (RSs) have become indispensable in shaping personalized experiences across a multitude of domains, profoundly influencing user choices in e-commerce, online streaming, web searches, and so forth [16, 22, 29, 67, 71]. RSs not only guide users through an overwhelming array of options but also drive engagement and customer satisfaction, making them critical to the success of digital platforms. Among the diverse techniques employed to decode complex user preferences, Graph Neural Networks (GNNs) [8, 38, 61] stand out as a groundbreaking approach. GNNs adeptly unravel the intricate patterns of user-item interactions, significantly enhancing the precision and effectiveness of recommendations [25, 70].

However, these methods deployed in the real world often underdeliver its promises made through the benchmark datasets [68]. This discrepancy largely stems from their traditional offline training and testing approach [35]. In these scenarios, models are trained on large, static datasets and then evaluated on limited test sets, a process that doesn't account for the dynamic nature of real-world data. In stark contrast, real-world RS are in a state of constant flux, where new user preferences, items, and interactions continually emerge, creates a gap that is essentially the difference in data distributions over time. On one hand, the models that are originally trained on historical data might not be well-equipped to handle such new, diverse data effectively. On the other hand, when these models are updated with the new data, they are at risk of overwriting the knowledge previously acquired—a phenomenon known as *Catastrophic Forgetting* [5]. This issue is notably problematic in RS, where retaining older but pertinent information is pivotal for sustaining a holistic grasp of user preferences and behavior. Consequently, although GNN-based RS models demonstrate considerable prowess, their ability to adapt to the perpetually changing data landscape poses a significant challenge requiring urgent and concentrated efforts.

Recent studies have aimed to embrace this challenge, and most of these works are delving into harnessing the potential of continual learning methods [7, 12, 34, 35, 43]. The first line of research [6, 21, 35, 41, 47] relies on a replay buffer to periodically retrain the model using a selection of past samples. However, the effectiveness of such sample-based methods diminishes with a reduced buffer size and becomes impractical in scenarios where using a replay buffer is constrained, such as in situations requiring strict data privacy [36]. This limitation is crucial: when the buffer fails to represent the complete spectrum of past data, the method struggles to preserve essential historical knowledge, leading to a gap between what is retained and the current data landscape.

The second line of works, model regularization-based methods [11, 18, 19, 45], aims to maintain knowledge by constraining the model's parameters to prevent significant divergence from previously learned configurations. These parameters are critical as they often encapsulate patterns extracted from historical data. Yet, the challenge arises when new data diverges substantially from these historical patterns. If the model's parameters are not adequately adaptable to this new information, it risks straying too far from relevant past data, triggering catastrophic forgetting. The last line of works relies on the model isolation and expansion strategies [15, 37, 42]. These strategies isolate old knowledge and create new learning spaces for updated data. Yet, their extensive model expansion often results in increased parameters and time-consuming updates. In essence, while these strategies appear promising, they fail to fully satisfy the key requirement of streaming recommendation: effectively bridging the ever-present gap between evolving new data and past data distributions. It's not just about preventing catastrophic forgetting but also about ensuring effective learning and adaptation to new data. This dual requirement is where these methods fall short, underscoring the need for more sophisticated strategies that can seamlessly integrate evolving data dynamics while retaining essential historical insights, thereby addressing the core challenge of streaming recommendation.

In order to tackle the aforementioned challenges, we draw inspiration from the concept of prompt tuning [30], a new transfer learning technique in the field of natural language processing (NLP). Intuitively, prompt tuning techniques reformulates learning downstream tasks from directly adapting model weights to designing prompts that "instruct" the model to perform tasks conditionally. A prompt encodes task-specific knowledge and has the ability to utilize pre-trained frozen models more effectively than ordinary fine-tuning [24, 44, 60]. This effectiveness stems from the prompts' ability to add contextual layers to the model's understanding, thereby adapting its responses to new data without altering the core model structure. Moreover, prompt tuning stands as a data-agnostic technique. Unlike methods that heavily rely on the data they were trained on, prompt tuning can navigate across different data distributions without being hindered by the gaps typically encountered in evolving datasets. This quality makes it immune to the pitfalls introduced by the gap in data distributions. Thus, prompt tuning emerges as a potent solution for meeting the dual requirements of continual learning in RS. It not only aids in preventing catastrophic forgetting by maintaining the integrity of the model's foundational knowledge but also ensures effective learning and adaptation to new and diverse data patterns.

In particular, applying prompt tuning in continual graph learning for recommender systems is challenging. First of all, existing prompt tuning approaches [20, 46, 50, 60, 65] are for data in Euclidean space, e.g., images, texts. In the dynamic user-item interaction graphs, however, changes are not just incremental; they are cascaded and interconnected, profoundly influencing the entire network. For example, the addition or removal of a node triggers a domino effect, altering the states of adjacent nodes and potentially leading to substantial changes across the entire graph. This cascading nature of change implies that incremental updates simultaneously affect multiple levels of relationships within the graph. Therefore, how to disentangle the multiple levels of relationship changes caused by the cascaded changing of graph data becomes a unique challenge for graph prompt tuning. Furthermore, once these relationships are disentangled, a significant task remains in re-aggregating these representations while preserving the overall graph's integrity. Efficiently combining these decoupled elements is vital for maintaining a coherent and accurate representation of the evolving graph. Last but not least, a major concern with existing methods is the lack of theoretical guarantees for the prompt tuning process in dynamic environments. This limitation can lead to unstable and inconsistent training, exacerbated by the continually changing nature of graph data in recommender systems.

To tackle the challenges above, in this paper, we introduce GPT4Rec, a **G**raph **P**rompt **T**uning method for streaming **Rec**ommendation. To address the challenge of managing cascaded changes, GPT4Rec disentangles the graph into distinct feature and structure views, which is optimized to capture unique characteristics of each semantic relationship within the graph. Central to GPT4Rec's design are three types of prompts tailored for specific aspects of graph data. Firstly, node-level prompts are employed to instruct the model to adapt to changes in the attributes or properties of individual nodes within the graph, ensuring a nuanced understanding of evolving node characteristics. Secondly, structure-level prompts guide the

model in adapting to broader patterns of connectivity and relationships within the graph, capturing the dynamic interplay between different graph elements. Finally, view-level prompts are innovatively designed to facilitate the aggregation of information from multiple disentangled views. This approach allows GPT4Rec to synthesize a comprehensive understanding of the graph, ensuring that all vital aspects of the user-item interactions are considered and effectively integrated. The utilization of lightweight graph prompts efficiently guides the model across varying interaction patterns within the user-item graph. This approach is in stark contrast to model-isolation methods, as GPT4Rec's prompt based strategy enables rapid adaptation to new data streams. It efficiently pinpoints and transfers useful components of prior knowledge to new data, thereby preventing the erasure of valuable insights. Finally, we provide theoretical analysis specifically from the perspective of graph data to justify the ability of our method. We theoretically show that GPT4Rec has at least the expression ability of fine-tuning globally using the whole data. Extensive experiments are conducted on six real-world datasets, where GPT4Rec outperforms state-of-the-art baselines for continual learning on dynamic graphs.

In summary, we make the following contributions:

- We propose GPT4Rec, a graph prompt tuning based approach tailored for streaming recommendation. By strategically utilizing node-level, structure-level, and view-level prompts, GPT4Rec effectively guides the model to recognize and incorporate new data trends without overwriting valuable historical knowledge.
- Theoretical analyses affirm that GPT4Rec has at least the expression ability of fine-tuning globally.
- We conduct extensive evaluations on four real-world datasets, where GPT4Rec achieves state-of-the-art on streaming recommendation.

2 RELATED WORK

2.1 Streaming Recommendation

Traditional recommender systems, constrained by static datasets, struggle with predicting shifting user preferences and trends due to the dynamic nature of user interactions and the expanding volume of items. Streaming recommendation, a dynamic approach updating both data and models over time, addresses these challenges [5, 7, 9, 10, 51, 52, 58]. While initial efforts focused on item popularity, recency, and trend analysis [4, 32, 53], recent advancements integrate collaborative filtering and matrix factorization into streaming contexts [5, 10, 12, 49]. Further, approaches using online clustering of bandits and collaborative filtering bandits have emerged [2, 13, 14, 26, 27]. The application of graph neural networks (GNNs) in streaming recommendation models is gaining attention for its complex relationship modeling [1, 56, 57, 69]. This shift towards streaming recommendation systems represents a significant advancement in the field, offering a more dynamic and responsive approach to user preference analysis and item suggestion.

2.2 Continual Learning

Continual learning (CL) addresses the sequential task processing with strategies to prevent catastrophic forgetting and enable knowledge transfer. The primary algorithms in continual learning are categorized into three groups: experience replay [6, 21, 35, 41, 47], model regularization [11, 18, 19, 45], and model isolation [15, 37, 42]. Recently, continual graph learning [3, 28, 33, 39, 40, 69] has emerged, focusing on chronological data in streaming recommendation systems [1, 56, 59, 62]. the focus shifts to handling data that arrives continuously in a chronological sequence, rather than data segmented by tasks. This research diverges from traditional continual learning by emphasizing effective knowledge transfer across time segments, instead of solely focusing on preventing catastrophic forgetting.

2.3 Graph Prompt tuning

Prompt tuning, a technique extensively employed in Natural Language Processing (NLP) and Computer Vision (CV), aims to bridge the gap between pre-training tasks and fine-tuning tasks. This methodology has recently seen increased application in graph-based scenarios, highlighting its versatility and effectiveness. In the realm of graph neural networks (GNNs), several innovative adaptations of prompt tuning have emerged: GPPT [54] leverages learnable graph label prompts, transforming the node classification task into a link prediction task to mitigate the task type gap. Graph-Prompt [31] introduces a universal prompt template to unify all the tasks via a learnable readout function. All-in-One [55] proposes an inventive graph token prompt, coupled with a token insertion strategy to align the pre-training and fine-tuning tasks. HGPrompt [66] extends the concept of prompt tuning to heterogeneous graphs by designing unique prompts for each node type. In the context of streaming recommendations, our work marks a pioneering effort in utilizing graph prompt tuning. We employ prompts to guide the model in swiftly adapting to new data streams, ensuring the continuous integration of evolving data while maintaining the integrity of previously learned information.

3 PRELIMINARIES

In this section, we first formalize the continual graph learning for streaming recommendation. Then we briefly introduce three classical graph convolution based recommendation models used in this paper.

3.1 Definitions and Formulations

Definition 1. Streaming Recommendation. Massive user-item interaction data $\tilde{D}$ streams into industrial recommender system continuously. For convenience [56, 57, 69], the continuous data stream is split into consecutive data segments $D_1, ..., D_t, ..., D_T$ with the same time span. At each time segment t, the model needs to optimize the recommendation performance on D_t with the knowledge inherited from $D_1, D_2, ..., D_{t-1}$. The recommendation performance is evaluated along the whole timeline.

Definition 2. Streaming Graph. A streaming graph is represented as a sequence of graphs $G = (G_1, G_2, ..., G_t, ...G_T)$, 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, respectively. $\Delta G_t = (\Delta A_t, \Delta X_t)$ is the changes of graph structures and node attributes at t. The changes contain newly added nodes and newly built connections between different nodes.

Definition 3. Continual Graph Learning for Streaming Graph.
Given a streaming graph $G = (G_1, G_2, ..., G_t, ...G_T)$, the goal of continual graph learning (CGL) is to learn $\Delta G_t(D_t)$ sequentially while transferring historical knowledge to new graph segments effectively. Mathematically, the goal of CGL for streaming graph is to find the optimal GNN structure S_t and parameters W_t at each segment t such that:

$$(S_t^*, W_t^*) = \underset{S_t, W_t}{\arg\min} \, \mathcal{L}_t(S_t, W_t, \Delta G_t), \qquad (1)$$

where $S_t, W_t \in (\mathcal{S}, \mathcal{W})$. $\mathcal{L}_t(S_t, W_t, \Delta G_t)$ is the loss function of current task defined on ΔG_t . The $\mathcal{S}$ and $\mathcal{W}$ are corresponding search spaces, respectively.

Since the user-item interaction data is actually a bipartite graph, the continual learning task for streaming recommendation is essentially the continual graph learning for streaming graph. For each segment t, the GNN structure S_t and parameters W_t need to be adjusted and refined simultaneously to achieve a satisfying recommendation performance. We use the Bayesian Personalized Ranking (BPR) [48] loss as the loss function in this work, because it is effective and has broad applicability in top-K recommendations.

4 METHODOLOGY

In this section, we introduce our GPT4Rec method towards continual graph learning for streaming recommendation. We first model the temporal user preference to capture the preference shifts between segments, which will be utilized for the user embedding initialization of training. Then, we successively use two operations, historical graph convolution pruning and refining as well as graph convolution expanding and pruning (shown in Figure 2), in such a model isolation way to obtain the best structure and optimal parameters of the graph convolution.

4.1 Disentangling Strategy for Complex Graphs

The user-item interaction graphs in recommender systems are inherently complex due to their dynamic and interconnected nature. When a new node is added or an existing one is removed, it doesn't just affect isolated parts of the graph. Instead, these changes can lead to cascading effects throughout the network. These cascading changes mean that any modification in the graph can simultaneously influence multiple relationships. For example, a new item added to the system could cultivate a new preference for several users, altering the existing user-item interaction patterns and potentially reshaping the overall landscape of preferences and recommendations. Such dynamics underscore the complexity of these graphs and the need for a modeling approach that can adapt to and capture these multifaceted and simultaneous changes.

In this context, we divide the graph into multiple views, where each view is tailored to capture specific aspects of the user-item interactions. The disentanglement is achieved through a series of linear transformations:

$$\tilde{x}_i = Linear_i(x), \qquad (2)$$

where $Linear_i$ is the linear transformation of the i-th view. The linear transformations are capable of separating these views while maintaining the overall integrity of the graph. This means the model can isolate and focus on specific aspects without losing sight of the graph's interconnected nature. This disentanglement allows the model to explore and identify distinct patterns and relationships in the data. For example, one view might capture user-to-item interactions, another might focus on user-to-user relationships, and yet another could delve into item-to-item similarities. With each view focusing on a particular aspect of the graph, the model can adapt more precisely to changes or updates in the data.

4.2 Prompt Design for Adaptive Learning

After disentangling the graph patterns into different views, we design node-level prompt and structure-level prompt to capture the comprehensive essence of graph patterns in dynamic RSs.

4.2.1 *Node-level Prompts.* The node-level prompts primarily target the attributes or properties of individual nodes within the graph. This could include user characteristics in a social network or item properties in a recommendation system. By focusing on this level, GPT4Rec can delve into the intricacies of node-specific data, allowing for a nuanced understanding of individual behaviors or item features. This focus is crucial for tasks where personalization or detailed attribute analysis is key.

Specifically, for each view, the node-level prompts is a set of learnable parameters $P = [p_1, ..., p_L]$, where L is the number of node-level prompts. These prompts act as targeted cues that inform the model about how to interpret and integrate new information regarding users or items.

Contextual Guidance Through Weighted Addition. When new data arrives, these prompts effectively 'instruct' the model by highlighting relevant features or changes in the user-item interactions:

$$\tilde{x}_i = \tilde{x}_i + \Sigma_j^L \alpha_{ij} p_j, \qquad (3)$$

$$\alpha_{ij} = \frac{exp(\tilde{x}_i p_j)}{\Sigma_r exp(\tilde{x}_i p_r)}, \qquad (4)$$

where α_{ij} is the weight for prompt j calculated based on its relevance to the current data point $\tilde{x}_i$. This is done using the softmax function, which essentially turns raw scores (obtained by multiplying x_i and p_j) into probabilities.

These prompts, each encoding specific patterns or relationships, are weighed differently for different nodes. This means that for a given node x_i, certain prompts will have higher weights (α_{ij}) if they are more relevant to that node's context. This selective amplification allows the model to focus on aspects of the data that are currently most pertinent. For instance, if a particular prompt encodes a pattern that is increasingly common in new data, its weight will be higher for nodes where this pattern is relevant.

As new data comes in, the relevance of different prompts can change. The model dynamically recalculates the weights (α_{ij}) for each node with each new data point, allowing it to adapt its focus continuously. This dynamic process ensures that the model remains responsive to evolving data trends and relationships, integrating new knowledge in a way that's informed by both the new and existing data.

4.2.2 *Structure-level Prompts.* Alongside the node-level prompts, structure-level prompts are designed to engage with the broader patterns of connectivity and relationship within the graph. These prompts are crucial for understanding and adapting to changes in
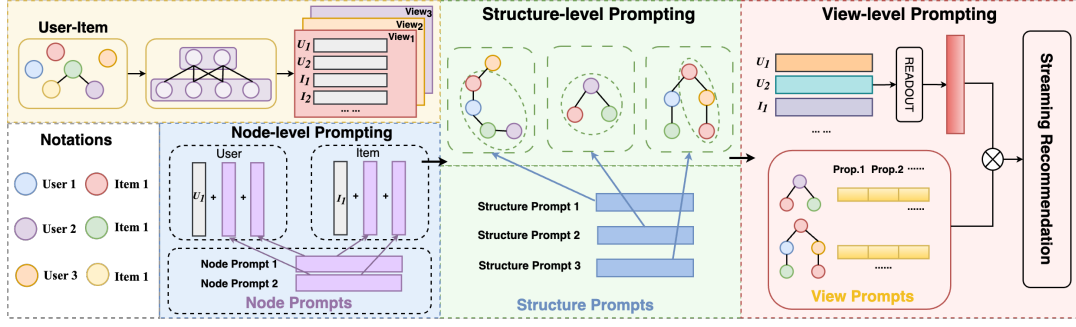
Figure 1: Overview of the GPT4Rec. a) We disentangle the complex user-item interaction graph into multiple views. b) We design the node-level prompt to depict the node-level changes. c) We propose the structure-level prompt for the decomposition of sub-graph structure. d) Finally, we present a view-level prompt to aggregate the disentangled views.

the overall graph topology, such as the emergence of new interaction patterns or the evolution of existing ones.

The structure-level prompt is designed as follows: for each view, we design a set of learnable prompt $Q = [q_1, ..., q_k]$ for the edges that adaptively aggregate the structure-level information via message-passing mechanism:

$$\tilde{x}_i = \tilde{x}_i + \Sigma_{j \in N(x_i)} \beta_{ij} u_{ij}, \tag{5}$$

$$\beta_{ij} = \frac{exp(\tilde{x}_i u_{ij})}{\Sigma_r \tilde{x}_i u_{ir}}, \tag{6}$$

$$u_{ij} = \Sigma_k^K att_{ij,k} q_k, \tag{7}$$

$$att_{ij,k} = \frac{exp(a(\tilde{x}_i || \tilde{x}_j) q_k)}{\Sigma_r exp(a(\tilde{x}_i || \tilde{x}_j) q_r)}, \tag{8}$$

where x_i and x_j are adjacent nodes, a is the linear mapping.

The integration of these prompts within each view facilitates a comprehensive and responsive learning process. By simultaneously addressing both the granular details at the node level and the broader structural dynamics, GPT4Rec ensures a holistic understanding of the graph data. This approach enables the model to effectively learn from and adapt to the continually evolving landscape of user-item interactions in streaming recommendation.

4.3 Aggregation of Disentangled Representations

The aggregation of information from multiple disentangled views is crucial for providing a comprehensive understanding of the dynamic and interconnected user-item interactions.

4.3.1 Initial strategy. One straightforward approach to this aggregation is the application of an attention mechanism. For the disentangled views, the aggregation can be formulated as:

$$\hat{x}_i = Atten(p(x_i), [\tilde{x}_{i,1}, ..., \tilde{x}_{i,n}]) = x_i + \Sigma_j \gamma_{i,j} \tilde{x}_{i,j}, \tag{9}$$

$$\gamma_{i,j} = \frac{exp(p(x_i) \tilde{x}_{i,j})}{\Sigma_r exp(p(x_i), \tilde{x}_{i,r})}, \tag{10}$$

where $p()$ is a linear transformation function. However, this approach may not fully account for the evolving nature of user-item interactions, especially with the introduction of new data. As the graph's structure changes, the relevance of different views and

their interrelations can shift, potentially rendering the existing fixed attention weights less effective.

4.3.2 Cross-view-level prompts for aggregation. To enhance the model's efficiency and adaptability in the face of these dynamic changes, GPT4Rec incorporates Cross-View-Level Prompts for dynamic adaptation. This approach centers on updating a small set of 'codebook' prompts $J = [j_1, ..., j_n]$, rather than relearning the entire model's parameters. These prompts serve as dynamic modifiers to the attention mechanism, allowing the model to adapt its focus efficiently:

$$\hat{x}_i = Atten(prompt(x_i), [\tilde{x}_{i,1}, ..., \tilde{x}_{i,n}]) = x_i + \Sigma_j \epsilon_{i,j} \tilde{x}_{i,j}, \tag{11}$$

$$\epsilon_{i,j} = \frac{exp(prompt(x_i) \tilde{x}_{i,j})}{\Sigma_r exp(prompt(x_i), \tilde{x}_{i,r})} = \frac{exp(p(x_i + j_i) \tilde{x}_{i,j})}{\Sigma_r exp(p(x_i + j_i), \tilde{x}_{i,r})}, \tag{12}$$

In this enhanced aggregation process, the prompts subtly adjust the attention weights, reflecting the current state and relationships within the graph. This strategy maintains the model's stability while enabling it to respond dynamically to new data, ensuring the final node embedding $\hat{x}$ remains relevant and accurate over time.

4.4 Discussions

In this section, we discuss the differences between graph prompt tuning, as employed in GPT4Rec, and traditional model-isolation-expansion methods [15, 37, 42], particularly addressing the nature of knowledge storage and adaptation in these approaches.differences between graph prompt tuning, as employed in GPT4Rec, and traditional model-isolation-expansion methods, particularly addressing your queries about the nature of knowledge storage and adaptation in these approaches. The core distinction lies in how each approach integrates new knowledge and preserves existing information.

Traditional model-isolation-expansion methods typically involve expanding the model's capacity to accommodate new information. This often means adding new layers or nodes, effectively increasing the model's size and complexity. While this approach can be effective in integrating new knowledge, it often requires significant resources and can lead to a bloated model. The expansion needs to be substantial enough to capture the new information, which might not be efficient or scalable in the long term. These methods can sometimes struggle with the delicate balance of preserving existing knowledge while incorporating new data. The expansion

can dilute the model's original understanding, potentially leading to issues like overfitting to recent data at the expense of older, yet still relevant, insights.

On the contrary, graph prompt tuning doesn't simply expand the model's space to store new knowledge. Instead, it introduces a set of contextually adaptive prompts that act as conduits for the new information. These prompts do not store knowledge in the traditional sense; they modify how the model interprets and processes incoming data. The prompts in GPT4Rec provide nuanced guidance, subtly adjusting the model's focus and understanding based on the current context. They act as dynamic, lightweight 'instructors' that align new data with the model's existing knowledge base. This is achieved without the need for extensive expansion of the model's structure, ensuring efficiency and agility. The key here is adaptability. The prompts are designed to be flexible, adjusting their influence based on the relevance to new data. This allows GPT4Rec to seamlessly integrate new insights while maintaining the integrity of previously learned information, thus avoiding catastrophic forgetting.

4.5 Theoretical Analysis

We conduct theoretical analysis to guarantee the correctness of the proposed graph prompt tuning algorithm on dynamic graphs. The conclusion is achieved by the following theorem with its proofs.

THEOREM 4.1. *GPT4Rec has at least the expression ability of fine-tuning globally using the whole data.*

PROOF. Suppose the model is updated at the time t, which means the model parameter θ_t is optimal. At time $t + 1$, the global fine-tuning process is as follows (taking x_i as the example):

$$\underset{\theta_{t+1}}{\arg\min} L(f_{\theta_{t+1}}(x_i^{t+1}), y_i^{t+1}), \tag{13}$$

where we use θ_t to initialize θ_{t+1} and L is the loss function. The optimization has the upper bound as[69]:

$$\underset{\theta^{t+1}}{\arg\min} L(f_{\theta_{t+1}}(\Delta x_i^t), y_i^{t+1}) + L(f_{\theta_t}(x_i^t), y_i^t), \tag{14}$$

where Δx_i^t denotes the node data gap, Equation 14 equals to Equation 13 when $\theta_{t+1} = 0$ and, representing the initialization process.

For the optimization of prompt, the optimization process is:

$$\underset{promp^{t+1}}{\arg\min} L(f_{\theta_t}(x_i^{t+1} + \Sigma_j \epsilon_{i,j} promp(x_i^{t+1})), y_i^{t+1}),$$

$$\Leftrightarrow \underset{promp^{t+1}}{\arg\min} L(f_{\theta_t}(\Delta x_i^{t+1} + \Sigma_j \epsilon_{i,j} promp^{t+1}(x_i^{t+1})+, y_i^{t+1})+ \tag{15}$$

$$L(f_{\theta^t}(x_i^t, y_i^{t+1}),$$

which is equivalent to we fix θ_t and optimize Equation 13. Namely, we optimize the gap from the optimal. Therefore, GPT4Rec has at least the expression ability of global fine-tuning with the whole data. □

5 EXPERIMENTS

In this section, we conduct experiments on four real-world time-stamped recommendation datasets to evaluate our proposal.

Table 1: Statistics of datasets used in the experiments.

Statistic	Tb2014	Tb2015	Netflix	Foursquare
No. of Users	8K	192K	301K	52K
No. of Items	39K	10K	9K	37K
No. of Interactions	749K	9M	49M	2M
Time Span	31 days	123 days	74 months	22 months
AER	35.5%	26.0%	58.4%	60.0%

5.1 Experiment Settings

5.1.1 Datasets. We conduct experiments on four datasets (*i.e.,* Netflix[1], Foursquare[2] [63, 64] and Taobao2014[3] and Taobao2015[4] from from Alibaba's M-Commerce platforms) from three different domains (i.e., social media, points-of-interests , and e-commerce). Following [17, 63, 64], we use the average entity overlapping rate (AER) between segments to assess data stream stability; higher AER indicates greater stability. Data in each segment is divided into training, validation, and test sets in an 8:1:1 ratio. The statistics of the four datasets are summarized in Table 1.

5.1.2 Baselines. We compare GPT4Rec with three types of baselines: (1) experience replay-based baselines, which includes Inverse Degree Sampling (Inverse) [1] and ContinualGNN [56]. (2) knowledge distillation-based baselines, which includes Topology-aware Weight Preserving (TWP) [28], GraphSAIL [62], SGCT [59], MGCT [59] and LWC-KD [59]. (3) parameter isolation-based baselines: DEGC [17]. (4) Vanilla Finetune baseline that initializes with parameters from the previous segment and fine-tunes using only the current segment's data.

5.1.3 Reproducibility. We apply grid search to find the optimal hyper-parameters for each model. The ranges of hyper-parameters are $\{32, 64, 96, 128\}$ for size L of node-level prompts P, the size K of structure-level prompts Q and the size N of cross-view-level prompts J. The range of disentangled view number is $\{2, 4, 8, 16\}$. Adam optimizer [23] is employed to minimize the training loss. Other parameters are tuned on the validation dataset and we save the checkpoint with the best validation performance as the final model. We use the same evaluation metrics **Recall@K** (abbreviated as **R@K**) and **NDCG@K** (abbreviated as **N@K**) following previous studies [17, 59, 62]. All models are run five times with different random seeds and reported the average on a single NVIDIA GeForce RTX 3090 GPU.

5.2 Comparison with Baseline Methods

In Table 2, we show the average performance of different methods on four datasets while choosing MGCCF as the base GCN recommendation model.

The traditional Finetune method, for instance, inherits and fine-tunes parameters from previous data segments. While this can be effective for incremental updates, it often leads to catastrophic forgetting when new learning overshadows previously acquired

---

[1]https://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a
[2]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[3]https://tianchi.aliyun.com/dataset/46
[4]https://tianchi.aliyun.com/dataset/dataDetail?dataId=53

Table 2: The average performance with MGCCF as our base model. The numbers in bold indicate statistically significant improvement (p < .01) by the pairwise t-test comparisons over the other baselines.

Dataset	Metric	(a) Finetune	(b) Uniform★	(c) Inverse★	(d) ContinualGNN*	(e) TWP[†]	(f) GraphSAIL[†]	(g) SGCT[†]	(h) MGCT[‡]	(i) LWC-KD	(j) DEGC	(k) GPT4Rec	(l) Improv.
Taobao2014	Recall@20	0.0412	0.0308	0.0323	0.0311	0.0398	0.0395	0.0423	0.0421	0.0440	<u>0.1082</u>	**0.1127**	+4.16%
	NDCG@20	0.0052	0.0038	0.0039	0.0035	0.0050	0.0051	0.0054	0.0055	0.0059	<u>0.0142</u>	**0.0149**	+4.93%
Taobao2015	Recall@20	0.4256	0.4194	0.4217	0.4203	0.4320	0.4371	0.4411	0.4446	0.4512	<u>0.4892</u>	**0.5018**	+2.58%
	NDCG@20	0.0117	0.0113	0.0114	0.0111	0.0122	0.0126	0.0129	0.0131	0.0139	<u>0.0167</u>	**0.0172**	+2.99%
Netflix	Recall@20	0.3359	0.3298	0.3321	0.3089	0.3428	<u>0.3470</u>	0.3300	0.3252	0.2616	0.2949	**0.3508**	+1.10%
	NDCG@20	0.0580	0.0533	0.0536	0.0491	0.0580	<u>0.0583</u>	0.0572	0.0562	0.0482	0.0520	**0.0589**	+1.03%
Foursquare	Recall@20	0.1154	0.1024	0.1063	0.1056	0.1048	0.1086	0.1255	0.1192	0.1280	<u>0.1425</u>	**0.1477**	+3.65%
	NDCG@20	0.0115	0.0101	0.0107	0.0098	0.0104	0.0110	0.0137	0.0127	0.0144	<u>0.0178</u>	**0.0185**	+3.93%

knowledge. GPT4Rec circumvents this issue through its adaptive integration of new data, preserving historical context alongside new insights. Experience replay methods like Uniform Sampling and Inverse Degree Sampling sample historical data to combine with new information. However, they may not always strike the right balance between old and new data, potentially missing nuanced changes in user-item interactions. GPT4Rec's prompt-based strategy offers a more precise response to evolving data patterns, ensuring a seamless blend of historical and current user preferences. Knowledge distillation and experience replay techniques employed in ContinualGNN, TWP, GraphSAIL, SGCT, MGCT, and LWC-KD focus on pattern consolidation. These methods can be effective but may not fully capture the dynamic nature of user-item interactions in streaming scenarios. GPT4Rec's dual-prompt strategy, responsive to both node-level and structure-level changes, provides a more granular understanding of evolving preferences. DEGC, which models temporal preferences and performs historical graph convolution pruning and expanding, is adept at isolating long-term preferences. However, it might not be as nimble in adapting to quick short-term shifts. In contrast, GPT4Rec's flexible framework allows for real-time adaptation to both long-term and immediate changes, offering a comprehensive view of user preferences.

GPT4Rec distinguishes itself by its efficient integration of new knowledge via prompts, making it both streamlined and resource-efficient. Moreover, GPT4Rec excels in maintaining a critical balance between preserving historical data and adapting to emerging trends. This balance is essential in dynamic streaming environments where both historical continuity and responsiveness to new patterns are necessary for accurate recommendations. The model achieves a more precise understanding of user-item relationships through its sophisticated use of node-level and structure-level prompts. These prompts enable GPT4Rec to adjust its recommendations based on the context of each interaction, considering both the individual characteristics of nodes (such as specific user preferences and item attributes) and the overall structure of the graph (such as the connectivity and clustering of nodes). This contextual sensitivity ensures that the model not only captures but also effectively interprets the

Table 3: The average performance with NGCF as our base model. The numbers in bold indicate statistically significant improvement (p < .01) by the pairwise t-test comparisons over the other baselines.

Model	Taobao2014		Netflix	
	R@20	N@20	R@20	N@20
Finetune	0.0304	0.0040	0.3131	0.0541
Uniform	0.0340	0.0038	<u>0.3263</u>	0.0525
Inverse	0.0347	0.0039	0.3256	0.0518
ContinualGNN	0.0338	0.0036	0.3047	0.0479
TWP	0.0358	0.0047	0.3159	0.0531
GraphSAIL	0.0318	0.0042	0.3245	<u>0.0554</u>
SGCT	0.0350	0.0046	0.3044	0.0533
MGCT	0.0346	0.0045	0.2957	0.0511
LWC-KD	0.0380	0.0050	0.2496	0.0454
DEGC	<u>0.0961</u>	<u>0.0123</u>	0.2713	0.0486
GPT4Rec	**0.0972**	**0.0129**	**0.3341**	**0.0573**

complex dynamics within the user-item graph, leading to more precise and relevant recommendations.

5.3 Method Robustness Analysis

To figure out whether our method is robust to different datasets and GCN-based recommendation models, we conduct the experiments with NGCF and LightGCN as the base GCN models on both Taobao2014 and Netflix datasets. The corresponding results are shown in Tables 3 and Table 4. For GCN models NGCF and Light-GCN, the improvements of our methods on Taobao2014 are both significant. GPT4Rec achieves the state-of-art recommendation performance. This also shows the performance potential of GPT4Rec on different kinds of base GCN models. As for the dataset Netflix, GPT4Rec improves the Recall@20 by 6.71% and NDCG@20 by 5.91% over Finetune when choosing NGCF as the GCN model. Similar improvements can also be observed when taking LightGCN as the base GCN model. Such observations demonstrate the robustness of our methods to different datasets and GNN backbones.

Table 4: The average performance with LightGCN as our base model. The numbers in bold indicate statistically significant improvement (p < .01) by the pairwise t-test comparisons over the other baselines.

| Model	Taobao2014		Netflix	
	R@20	N@20	R@20	N@20
Finetune	0.0339	0.0040	0.3179	0.0537
Uniform	0.0377	0.0041	0.3289	0.0533
Inverse	0.0386	0.0042	0.3275	0.0530
ContinualGNN	0.0382	0.0041	0.3035	0.0475
TWP	0.0338	0.0040	0.3204	0.0542
GraphSAIL	0.0342	0.0042	0.3282	0.0544
SGCT	0.0342	0.0043	0.3073	0.0519
MGCT	0.0357	0.0047	0.2983	0.0516
LWC-KD	0.0402	0.0053	0.2571	0.0461
DEGC	0.0975	0.0125	0.2776	0.0491
GPT4Rec	**0.0979**	**0.0131**	**0.3348**	**0.0567**

The underlying reasons for these improvements and the robust nature of GPT4Rec can be attributed to several factors. Firstly, the model's unique prompt-based approach allows for a more context-aware adaptation to evolving user preferences and item characteristics. This approach ensures that the recommendations are not only accurate but also relevant to the current data landscape. Secondly, GPT4Rec's ability to dynamically integrate new information while preserving valuable historical insights helps maintain a balance that is crucial for the accuracy and relevance of recommendations in continually changing environments. Moreover, GPT4Rec's flexible framework adapts effectively to the inherent characteristics of different GCN models. Whether it's the complicated user-item interaction modeling in NGCF or the simplified yet efficient structure of LightGCN, GPT4Rec enhances these base models by effectively addressing their limitations and capitalizing on their strengths.
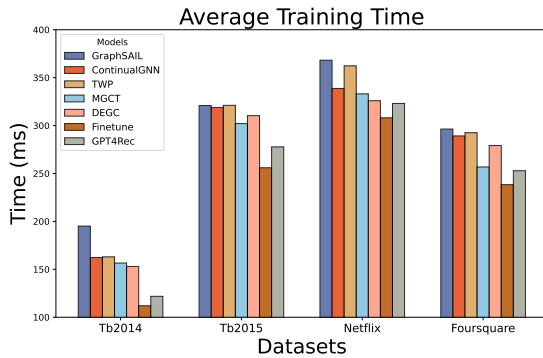
Figure 2: Average training time per epoch

5.4 Efficiency

Figure 2 presents the average training time of GPT4Rec per epoch compared with baselines. From the results, we observe that GPT4Rec not only aligns closely with the efficiency of the Finetune approach

(a) Recall@20

(b) NDCG@20
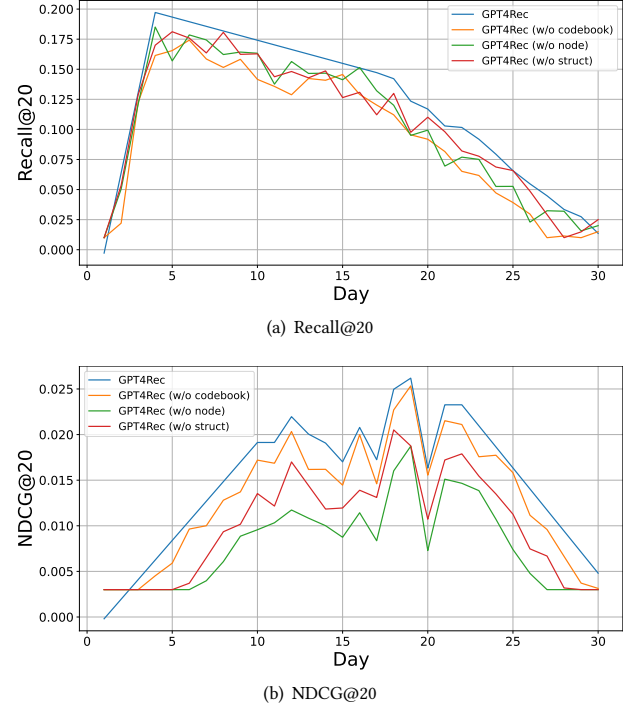
Figure 3: Ablation study on Taobao2014 dataset.

but also surpasses several other advanced models in terms of training speed. The high efficiency of GPT4Rec can be primarily attributed to the utilization of lightweight graph prompts. These prompts, despite their minimal computational footprint, play a crucial role in seamlessly integrating new data into the model. By leveraging these compact yet effective prompts, GPT4Rec bypasses the need for extensive retraining or large-scale parameter adjustments typically required by other models. This light-weight approach ensures rapid adaptability to new information, significantly reducing the computational overhead and training time.

5.5 Ablation Study

In this section, we focus on GPT4Rec and test the efficacy of its various designs in regard to the view disentangle, node-level prompt, structure-level prompt and view aggregation. The results are shown in Figure 3. We have the following observations:

- The process of decomposing the graph into distinct views is a fundamental aspect of the GPT4Rec model. This disentanglement facilitates the model's understanding by allowing it to separately process diverse interaction dynamics between users and items. The results suggest that disentangling the graph into multiple views substantially improves the model's capability to accurately represent and interpret the multifaceted relationship dynamics. Specifically, the disentangled views enable the model to address different aspects of the graph's structure in isolation, thus enhancing the quality of its recommendations by providing a more precise understanding of the interaction patterns.

(a) Recall@20   (b) NDCG@20
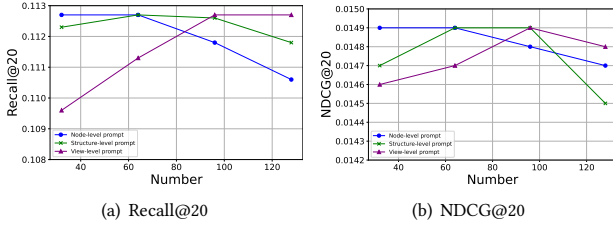
Figure 4: Influence of Prompt Number on Taobao2014 dataset.

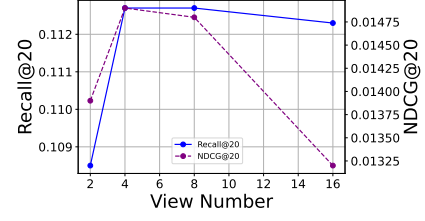**Figure 5: Influence of View Number on Taobao2014 dataset.**

- We evaluate the specific contribution of node-level prompts on adapting GPT4Rec to changes in individual nodes. By comparing versions of GPT4Rec with and without these prompts, we observe significant improvements in the model's response to shifts in user preferences and item attributes. These findings highlight the prompts' role in providing context-specific adjustments to the model, enhancing its ability to personalize recommendations.
- The effectiveness of structure-level prompts is analyzed by assessing how well GPT4Rec adapts to overall structural changes in the graph. The comparison reveals that including structure-level prompts leads to a more accurate representation of the global interaction patterns, demonstrating their importance in capturing broader relationship dynamics in the graph.
- Finally, the view aggregation component is scrutinized. This aspect is crucial for reintegrating the disentangled views into a cohesive model output. The ablation study shows that effective aggregation is key to ensuring that the insights gained from the separate views are synergistically combined, leading to a more comprehensive understanding of the graph data.

5.6 Hyperparameter Study

We conduct detailed hyperparameter studies on the hyperparameter of our model.

5.6.1 Prompt Size Study. We first study the impact of varying the size of node-level prompts P with size L, structure-level prompts Q with size K, and cross-view-level prompts J with size N on the Tb2014 dataset. The results are shown in Figure 4. We find that initially increasing the size of these prompts generally enhances the performance, indicating that Larger prompt sizes allow for a richer representation of complex user-item interactions and relationships, providing the model with a more diverse set of "hints" to interpret and integrate new information effectively. When the prompt size is large enough, we notice that further increasing the prompt size only brings marginal benefits. Moreover, excessively large prompts introduce additional computational overhead, which may not be justifiable given the marginal gains in performance.

Interestingly, the optimal size of prompts also varies across different types: cross-view-level prompts require a larger size, while node-level prompts are most effective with a smaller size. This variation can be explained by considering the nature of changes each prompt type addresses. Cross-view-level prompts need to capture broader and more complex patterns of interaction across different graph views, which may necessitate a larger size for comprehensive

representation. In contrast, node-level prompts, which target more specific and localized information, can achieve optimal performance with a smaller set of parameters.

5.6.2 View Size Study. We explore the influence of the number of disentangled views on the Tb2014 dataset. The results are shown in Figure 5. We find that an increase in the number of disentangled views generally corresponds to improved performance, primarily due to each view providing a distinct lens through which the model can perceive and process various aspects of user-item interactions. This multi-view approach enables a richer, more layered understanding of the data, as each view contributes unique insights into different facets of the graph, such as varying user preferences or item characteristics.

However, our findings also highlight a critical threshold beyond which additional views may begin to hinder rather than help the model's performance. When the number of views crosses this threshold, the model encounters challenges in synthesizing and harmonizing these diverse perspectives. Additionally, managing an excessive number of views introduces significant computational challenges. The increased complexity not only escalates the computational costs but also amplifies the risk of the model overfitting. This issue underscores the importance of finding an optimal balance in the number of views, where the model can benefit from diverse perspectives without being overwhelmed by them or incurring prohibitive computational expenses.

6 CONCLUSION

In this paper, we propose GPT4Rec, a graph prompt tuning method for the continual learning in recommender systems. We propose to decouple the complex user-item interaction graphs into multiple semantic views, which enables the model to capture a wide range of interactions and preferences. The use of linear transformations in this decoupling process ensures that each view is distinctly represented while maintaining the overall structural integrity of the graph. The introduction of node-level, structure-level, and cross-view-level prompts in GPT4Rec is a significant methodological advancement. These prompts serve as dynamic and adaptive elements within the model, guiding the learning process and ensuring that the model remains responsive to new and evolving patterns within the graph. Extensive experiments validate our proposal.

ACKNOWLEDGEMENTS

REFERENCES

[1] Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. 2021. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2832–2836.

[2] Yikun Ban and Jingrui He. 2021. Local clustering in contextual multi-armed bandits. In *Proceedings of the Web Conference 2021*. 2335–2346.

[3] Jie Cai, Xin Wang, Chaoyu Guan, Yateng Tang, Jin Xu, Bin Zhong, and Wenwu Zhu. 2022. Multimodal continual graph learning with neural architecture search. In *Proceedings of the ACM Web Conference 2022*. 1292–1300.

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

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

[6] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. 2019. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*.

[7] Chen Chen, Hongzhi Yin, Junjie Yao, and Bin Cui. 2013. Terec: A temporal recommender system over tweet stream. *Proceedings of the VLDB Endowment* 6, 12 (2013), 1254–1257.

[8] Tianwen Chen and Raymond Chi-Wing Wong. 2020. 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*. 1172–1180.

[9] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. 271–280.

[10] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 189–198.

[11] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5138–5146.

[12] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. In *Proceedings of the sixth ACM conference on Recommender systems*. 59–66.

[13] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *International Conference on machine learning*. PMLR, 1253–1262.

[14] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online clustering of bandits. In *International conference on machine learning*. PMLR, 757–765.

[15] Siavash Golkar, Michael Kagan, and Kyunghyun Cho. 2019. Continual learning via neural pruning. *arXiv preprint arXiv:1903.04476* (2019).

[16] Jiayan Guo, Peiyan Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. 2022. Evolutionary Preference Learning via Graph Nested GRU ODE for Session-based Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 624–634.

[17] Bowei He, Xu He, Yingxue Zhang, Ruiming Tang, and Chen Ma. 2023. Dynamically Expandable Graph Convolution for Streaming Recommendation. *arXiv preprint arXiv:2303.11700* (2023).

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[19] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 831–839.

[20] Zhiyuan Hu, Jiancheng Lyu, Dashan Gao, and Nuno Vasconcelos. 2023. POP: Prompt Of Prompts for Continual Learning. *arXiv preprint arXiv:2306.08200* (2023).

[21] David Isele and Akansel Cosgun. 2018. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[22] Yiqiao Jin, Yunsheng Bai, Yanqiao Zhu, Yizhou Sun, and Wei Wang. 2022. Code Recommendation for Open Source Software Developers. In *Proceedings of the ACM Web Conference 2023*.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[24] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[25] Chaozhuo Li, Senzhang Wang, Yukun Wang, Philip Yu, Yanbo Liang, Yun Liu, and Zhoujun Li. 2019. Adversarial learning for weakly-supervised social network alignment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 996–1003.

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

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

[28] Huihui Liu, Yiding Yang, and Xinchao Wang. 2021. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 8653–8661.

[29] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149* (2023).

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

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

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

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

[34] Fei Mi and Boi Faltings. 2020. Memory augmented neural model for incremental session-based recommendation. *arXiv preprint arXiv:2005.01573* (2020).

[35] Fei Mi, Xiaoyu Lin, and Boi Faltings. 2020. Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 408–413.

[36] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. 2020. Recommender systems and their ethical challenges. *Ai & Society* 35 (2020), 957–967.

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

[38] Z. Pan, Fei Cai, Wanyu Chen, Honghui Chen, and M. Rijke. 2020. Star Graph Neural Networks for Session-based Recommendation. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020).

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

[40] Massimo Perini, Giorgia Ramponi, Paris Carbone, and Vasiliki Kalavri. 2022. Learning on streaming graphs with experience replay. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. 470–478.

[41] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 524–540.

[42] Qi Qin, Wenpeng Hu, Han Peng, Dongyan Zhao, and Bing Liu. 2021. Bns: Building network structures dynamically for continual learning. *Advances in Neural Information Processing Systems* 34 (2021), 20608–20620.

[43] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 669–678.

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[45] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *Proceedings of the IEEE international conference on computer vision*. 1320–1328.

[46] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314* (2023).

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[48] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[49] Steffen Rendle and Lars Schmidt-Thieme. 2008. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*. 251–258.

[50] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 2023. CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11909–11919.

[51] Qingquan Song, Xiao Huang, Hancheng Ge, James Caverlee, and Xia Hu. 2017. Multi-aspect streaming tensor completion. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining.* 435–443.

[52] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-Chien Lee, and C Lee Giles. 2008. Real-time automatic tag recommendation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* 515–522.

[53] Karthik Subbian, Charu Aggarwal, and Kshiteesh Hegde. 2016. Recommendations for streaming data. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* 2185–2190.

[54] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 1717–1727.

[55] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in One: Multi-Task Prompting for Graph Neural Networks. (2023).

[56] Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. 2020. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 1515–1524.

[57] Junshan Wang, Wenhao Zhu, Guojie Song, and Liang Wang. 2022. Streaming graph neural networks with generative replay. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 1878–1888.

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

[59] Yuening Wang, Yingxue Zhang, and Mark Coates. 2021. Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 3518–3522.

[60] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 139–149.

[61] S. Wu, Y. Tang, Yanqiao Zhu, L. Wang, X. Xie, and T. Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *AAAI.*

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

[63] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *The world wide web conference.* 2147–2157.

[64] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudré-Mauroux. 2020. Lbsn2vec++: Heterogeneous hypergraph embedding for location-based social networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1843–1855.

[65] Zaiquan Yang, Yuqi Zhang, Yuxin Du, and Chao Tong. 2022. Semantic-aligned reinforced attention model for zero-shot learning. *Image and Vision Computing* 128 (2022), 104586.

[66] Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. 2023. HGPROMPT: Bridging Homogeneous and Heterogeneous Graphs for Few-shot Prompt Learning. *ArXiv* abs/2312.01878 (2023). https://api.semanticscholar.org/CorpusID: 265609471

[67] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining.* 168–176.

[68] Peiyan Zhang and Sunghun Kim. 2023. A Survey on Incremental Update for Neural Recommender Systems. *arXiv preprint arXiv:2303.02851* (2023).

[69] Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. 2023. Continual Learning on Dynamic Graphs via Parameter Isolation. *arXiv preprint arXiv:2305.13825* (2023).

[70] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709* (2022).

[71] Peilin Zhou, Meng Cao, You-Liang Huang, Qichen Ye, Peiyan Zhang, Junling Liu, Yueqi Xie, Yining Hua, and Jaeboum Kim. 2023. Exploring recommendation capabilities of gpt-4v (ision): A preliminary case study. *arXiv preprint arXiv:2311.04199* (2023).