Neural Inductive Biases for Sequential Recommendation

Chaoyue He

School of Computer Science & Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

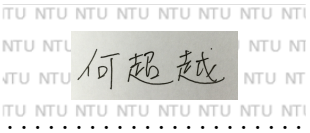
2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

30/Nov/2023

Date



Chaoyue He

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

30/Nov/2023

Date

Prof. Miao Chunyan

Authorship Attribution Statement

This thesis contains material from 1 paper accepted in the following peer-reviewed conference and 1 paper submitted to the following peer-reviewed journal in which I am listed as the first author.

Chapter 3 is accepted as Chaoyue He, Yong Liu, Qingyu Guo and Chunyan Miao, "Multi-Scale Quasi-RNN for Next Item Recommendation", in *Proceedings of the 8th International Conference on Crowd Science and Engineering (ICCSE)*, 2025.

The contributions of the co-authors are as follows:

- I drafted the manuscript and implemented the proposed method. I also conducted the experiments and analyzed the results.
- Dr. Liu and Dr. Guo provided valuable suggestions on the manuscript.
- Prof. Miao provided guidance on the research direction and manuscript writing.

Chapter 4 is submitted as Chaoyue He and Chunyan Miao, "Transitional Graph Attention Network for Next Basket Recommendation". Submitted to the International Journal of Crowd Science.

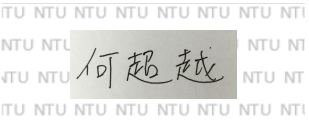
The contributions of the co-authors are as follows:

- I drafted the manuscript and implemented the proposed method. I also conducted the experiments and analyzed the results.
- Prof. Miao provided guidance on the research direction and manuscript writing.

30/NOV/2023

.....

Date



.....

Chaoyue He

Acknowledgements

I wish to express my special thanks to my supervisor, Prof. Miao Chunyan, for her full support throughout my entire Ph.D. journey.

I would like to extend my greatest gratitude to my senior and friend, Dr. Tay Yi. His deep insights and inspirations have been a guiding light not only in my research but also in life.

I am also grateful to Prof. Huang Mingting for her proofreading and valuable suggestions to improve the quality of my thesis.

I am deeply thankful to my colleagues at LILY. Mr. Huang Bo's continuous technical support, Dr. Zeng Zhiwei's guidance on the presentation rehearsals and in improving the quality of my thesis, Mrs. Hana's relentless administrative assistance, and Dr. Zhang Yinan's help in communications have all been crucial to my success. I also want to acknowledge all others in the center who have supported me over the past years.

My sincere appreciation goes to my seniors and the professors who discussed with me in the early stage of my research and provided valuable insights: Dr. Guo Qingyu, Dr. Liu Yong, Prof. Yu Han, Prof. Sun Aixin, Prof. Wang Ke, and Prof. Li Boyang.

Lastly, but certainly not least, I wish to express my heartfelt gratitude to my family for their unconditional love and support. My loved ones and friends have also been a source of continuous encouragement and support. Additionally, I am grateful to all the kind strangers and influential individuals who have provided me with warmth and inspiration in my life.

Abstract

In today's data-driven world, recommendation systems are indispensable for personalizing content to user preferences. Within this field, **sequential recommendation**—focused on ordered or timestamped user interactions—plays an essential role in anticipating future actions.

Despite significant advancements in this field, three key challenges remain underexplored: modeling variable-length user histories, capturing complex relationships in item sets, and addressing asynchronous temporal dynamics. Firstly, a significant limitation is the reliance on fixed-length user histories, which neglects the potential benefits of multi-scale or variable-length user subsequence interactions. This oversight hinders the effective capture of nuanced dynamics inherent in user interactions, ultimately creating a notable gap in the adaptability and accuracy of recommendations. Secondly, when transitioning from recommending single items to sets or baskets, the complexity of item relationships increases significantly. Contemporary systems struggle to effectively map and represent these intricate connections, underscoring a critical need for advanced methodologies to unravel and comprehend these multifaceted item relationships. These challenges are further compounded by the third gap - the mishandling of the temporal dimension. Real-world scenarios, characterized by asynchronous timestamps in user-item interactions, necessitate a refined approach. Existing methods, constrained by either solely focusing on absolute timestamps or limited to relative time intervals, fall short of capturing the intricate temporal dynamics that interweave with both variable-length sequences and complex item relationships.

To address these interconnected challenges, this thesis investigates a central research question: **How can neural inductive biases be developed to enhance sequential recommendation systems by effectively modeling variable interaction lengths, complex item relationships, and temporal dynamics?** The thesis systematically examines the often-overlooked sequential dimensions of

recommendation systems, particularly focusing on the order and timing of user actions. At the core of our approach is the implementation of innovatively designed neural inductive biases - carefully crafted assumptions embedded within neural architectures that guide the learning process toward more effective sequential recommendations.

The thesis presents a progressive exploration of solutions, beginning with **next-item recommendation** through a multi-scale-based approach. By incorporating specialized convolution within RNN cells, this method captures user-item interactions at different scales of orders, effectively addressing the challenge of modeling variable-length sequences and demonstrating improved recommendation accuracy.

Building upon this foundation, the thesis advances to **next-basket recommendation**, proposing a graph-attention-based method that models transitions between items across consecutive baskets. This approach solves basket-level recommendation by tackling more complex item relationships, utilizing an attention mechanism to capture intricate item dependencies and showing strong performance.

Further evolving the research, the thesis progresses to **next-items recommendation** by integrating asynchronous timestamps and multi-scale interaction utilizing a transformer architecture. This solution combines insights from previous approaches while implementing temporal warping embedding on timestamp intervals within a transformer architecture enriched with multi-scale convolution, effectively handling both variable interaction lengths and temporal dynamics.

Lastly, the thesis explores **temporal sets recommendation** through an innovative nested transformer architecture that simultaneously models inner-set item relationships and outer-set temporal dependencies. This final approach incorporates temporal quantization to convert continuous timestamps into discrete intervals and maps them to learnable encoding vectors. By leveraging a nested transformer structure, the model effectively addresses the dual challenges of complex item relationships and temporal dynamics in a unified and principled manner.

Looking beyond the current research, the thesis concludes by identifying promising research directions that build upon this exploration. These directions aim to advance the field's understanding of variable-length sequences, complex item relationships, and temporal dynamics in sequential recommendation systems, while exploring emerging challenges as recommendation systems continue to evolve.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

In an era characterized by an overwhelming abundance of information, recommendation systems have become indispensable tools for filtering relevant content and enhancing user experience. These systems help users navigate through vast amounts of data by providing personalized suggestions tailored to their preferences and needs. According to Grand View Research[1], as illustrated in Figure 1.1, the recommendation engine market was valued at USD 3.92 billion in 2023 and is projected to expand rapidly with a compound annual growth rate (CAGR) of 36.30% from 2024 to 2030. This remarkable growth underscores the critical role of recommendation engines as businesses increasingly prioritize personalized customer experiences and operational efficiency. These sophisticated engines harness advanced algorithms, particularly machine learning (ML) and artificial intelligence (AI), to process vast quantities of data and deliver tailored recommendations that precisely match individual user preferences.

Traditional recommendation systems often operate under the assumption of static user preferences, overlooking the sequential and temporal dynamics of interaction data. In contrast, sequential recommendation represents a significant advancement by modeling user interactions as a sequence of entities (e.g., items, baskets, or temporal sets) and focusing on predicting users' next likely interactions in the

[1] https://www.grandviewresearch.com/industry-analysis/recommendation-engine-market-report

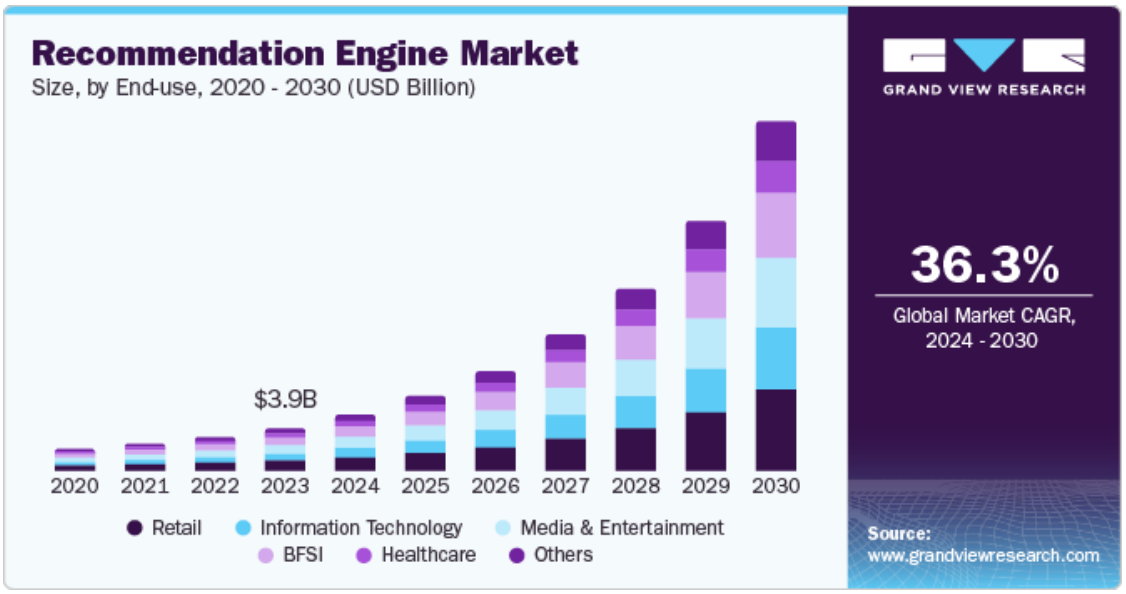


FIGURE 1.1: Global Recommendation Market Size

near future. This paradigm shift recognizes that user preferences are dynamic and evolving, with naturally existing order and timing dependencies that can be leveraged to enhance recommendation accuracy. By incorporating these temporal dynamics, particularly evident in domains such as purchase logs and viewing history, sequential recommendation systems deliver more accurate and relevant suggestions tailored to users' changing preferences. While Markov Chain-based models [1–5] pioneered sequential modeling, their limitations in capturing complex dynamic user preferences became evident. The emergence of deep learning, which has revolutionized fields like computer vision and natural language processing, is now transforming sequential recommendation systems. The quest for effective architectures to represent sequential and temporal user interactions has spawned diverse neural inductive biases, including recurrence[6–10], convolution[11–15], self-attention[16–20], and graph networks[21–25]. These approaches provide various ways to leverage the inherent order and timing information present in user behaviors, enabling more nuanced and accurate recommendations.

1.2 Objectives

This thesis addresses three fundamental challenges in current sequential recommendation systems:

TABLE 1.1: Summary of Sequential Recommendation Tasks and Their Formulations

Task	**Formulation**
Next Item Recommendation	Given each user's history record $L^u = \{x_1^u, x_2^u, \ldots, x_{T_u}^u\}$, predict the next item $x_{T_u+1}^u$.
Next Basket Recommendation	Given each user's history record $B^u = \{B_1^u, B_2^u, \ldots, B_{T_u}^u\}$, predict the next basket of items $B_{T_u+1}^u$.
Next Items Recommendation (Seq2Seq)	Given each user's history record $L^u = \{x_1^u, x_2^u, \ldots, x_{T_u}^u\}$, predict the next items $x_{T_u+1}^u, x_{T_u+2}^u, x_{T_u+3}^u, \ldots$.
Temporal Sets Recommendation (Seq2Seq)	Given each user's set of items history record $S^u = \{S_1^u, S_2^u, \ldots, S_{T_u}^u\}$ and corresponding timestamps $T^u = \{t_1^u, t_2^u, \ldots, t_{T_u}^u\}$, predict the next sets of items $S_{T_u+1}^u$ (or with the given timestamp $t_{T_u+1}^u$), $S_{T_u+2}^u$ (or with the given timestamp $t_{T_u+2}^u$) $\ldots$.

1. **Fixed-Length Single-Scale User History Limitation:** Current approaches predominantly rely on fixed-length single-scale user histories, which significantly constrains their analytical capabilities. Traditional methods such as FPMC[1] and Fossil[2] are limited to handling only first-order transitions between items. Similarly, RNN-based approaches like GRU4Rec[6] and CNN-based models like Caser[11] operate on fixed sequence lengths, imposing artificial constraints on the interaction history. This rigid design framework fails to adapt to the multi-faceted nature of user-item interactions, resulting in an inability to capture the full spectrum of behavioral patterns and preference dynamics that exist at various order scales.

2. **Complexity in Items Relationships Representation:** Moving from single items to baskets or sets introduces significant challenges in modeling complex item relationships. Current methods exhibit notable limitations in this area. For instance, HRM[26] employs only direct item combinations without considering deeper relationships, while NNRec[27] relies on simplistic feed-forward networks that cannot capture nuanced item interactions. Even more advanced approaches like DREAM[7] and Sets2Sets[28] are limited to basic

recurrent transitions between sets, failing to model the intricate dependencies between items within and across baskets of sets. There is a clear need for more sophisticated approaches that can effectively capture and represent the complex, multi-dimensional relationships between items in basket-level or set-based recommendations.

3. **Handling More Comprehensive Temporal Dynamics:** Time plays a crucial role that is inadequately addressed by conventional methods. Current approaches like attention-based SASRec[16] and ANAM[29] primarily focus on sequential order while neglecting actual temporal information between interactions. Even advanced transformer-based approaches like Transformer[30] and TSNDSP[31] are limited by their reliance on simple positional encoding, which fails to capture meaningful relative temporal relationships and intervals between interactions. There is a critical need for more sophisticated temporal modeling approaches that can effectively capture both the sequential nature and temporal dynamics of user-item interactions, including irregular time intervals and evolving user preferences over time.

Based on these challenges, this thesis aims to answer the central research question: **How can neural inductive biases be developed to enhance sequential recommendation systems by effectively modeling variable interaction lengths, complex item relationships, and temporal dynamics?**

To address these challenges, this thesis focuses on developing neural inductive biases for four key sequential recommendation tasks: (i) Next Item Recommendation, which predicts a single next item; (ii) Next Basket Recommendation, which predicts a set of items to be purchased together; (iii) Next (Sequence of) Items Recommendation, which predicts multiple future items in sequence; and (iv) Temporal Sets Recommendation, which incorporates timing information to predict future sets of items. Table 1.1 presents the detailed mathematical formulations of these tasks, providing a foundation for the comprehensive research approach and contributions of this thesis in advancing sequential recommendation systems.

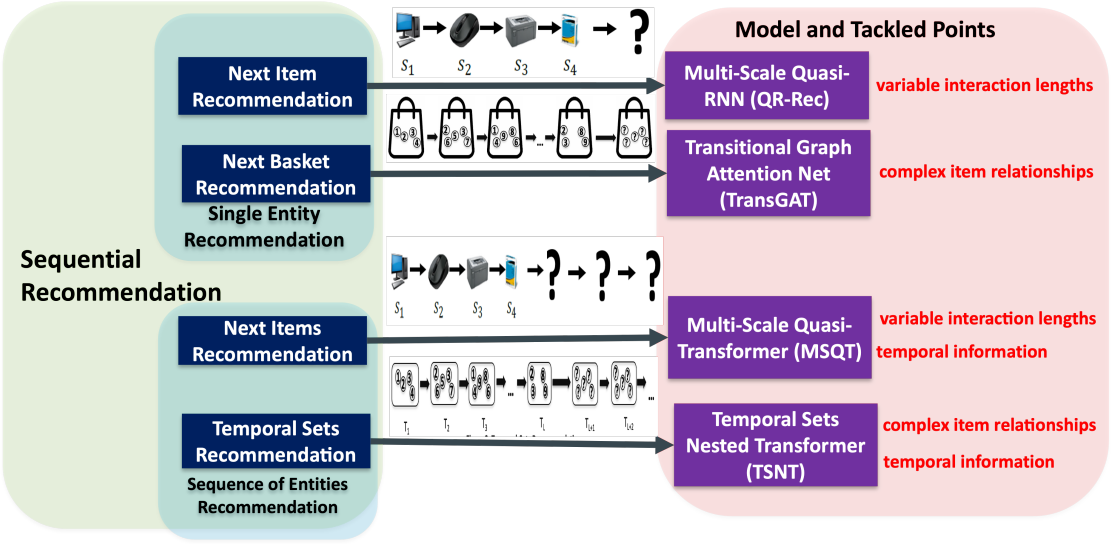


FIGURE 1.2: Sequential Recommendation Tasks and Solutions

1.3 Contributions

This thesis presents several significant contributions to the field of sequential recommendation. Beginning with a comprehensive literature review from an inductive bias perspective, it provides essential background and theoretical foundations to develop novel neural inductive biases specifically designed to address key challenges in sequential recommendation tasks. The review synthesizes and analyzes existing approaches while identifying critical gaps in current methodologies.

Beyond the literature review, the thesis makes four major technical contributions addressing distinct sequential recommendation tasks, as illustrated in Figure 1.2. These contributions systematically tackle the challenges of variable interaction lengths, complex item relationships, and temporal dynamics through innovative neural architectures and methodologies. The specific details and technical innovations of each contribution are elaborated in the following subsections.

1.3.1 Sequential Recommendation Literature Review

This literature review presents a comprehensive analysis of inductive biases throughout the sequential recommendation pipeline, with particular emphasis on neural inductive biases emerging from neural architectures and their crucial role in shaping system design and performance. The review systematically examines how these

biases influence different components and stages of recommendation systems, providing insights into their impact on overall effectiveness.

- This work presents comprehensive statistics spanning over eight years, encompassing approximately 530 research works. It highlights trends and progress from various top-tier conferences and journals in the field of sequential recommendation.
- A unified definition and formulation for various types of sequential recommendation setups is introduced, integrating previously separated definitions.
- A novel taxonomy based on the various stages of inductive bias is derived to summarize and categorize existing works, emphasizing diverse neural inductive biases discussions. This taxonomy offers valuable insights into the design choices of different models and suggests more effective strategies for sequential recommendation.

1.3.2 Multi-Scale Quasi-RNN (QR-Rec)

QR-Rec addresses the first aspect of the **central research question** by developing neural inductive biases to handle variable interaction lengths in sequential recommendation. Building on the extension of Quasi-RNN, a novel neural architecture proven effective for capturing milti-scale dependencies in sequential data, this model focuses on enhancing single item recommendation precision. Specifically, it tackles the research question: *How to model item order information beyond a single fixed-length user interaction sequence for next-item recommendation?* QR-Rec's key contributions include:

- **Extension of Quasi-RNN with Multi-Scale Framework:** Building on the foundation of Quasi-RNN, QR-Rec integrates dynamic average pooling to construct a multi-scale framework. This design enables the model to utilize convolutional filters of varying scales for processing user-item interaction sequences. By capturing different union-level features and feeding them into a recurrent structure, QR-Rec enhances sequential representation learning, offering a more nuanced understanding of user-item interactions. This approach directly addresses the first research question by leveraging multi-scale

user-item interactions to improve the precision of recommending the next single item.

- **Hierarchical Summation Aggregation:** The Hierarchical Summation Aggregation (HSA) strategy is introduced to optimize the aggregation of outputs across the model's architecture. This approach ensures an even contribution from each unit cell and component, maintaining the integrity of the original information without scaling. The HSA strategy has proven to be effective in maximizing the performance of QR-Rec.
- **Empirical Validation on Diverse Datasets:** QR-Rec's efficacy is validated through experiments on 15 real-world Amazon Reviews datasets, each with its unique distribution. The model consistently outperforms state-of-the-art methods across three key metrics. Detailed analyses of the model's hyperparameters and components have been conducted to understand the factors driving its superior performance.

1.3.3 Transitional Graph Attention Net (TransGAT)

TransGAT addresses the second aspect of the **central research question** by developing neural inductive biases to capture transitional relations between items in sequential recommendation. This model offers a novel neural architecture tailored for next-basket recommendation by addressing the research question: *How to represent basket items relationships in the transitional sequences effectively for next basket recommendation?* Its key contributions include:

- **Novel Graph Attention Architecture:** TransGAT pioneers the integration of graph attention networks in next-basket recommendation by introducing an innovative architecture centered around transitional graphs. As the first model to incorporate GAT for capturing item transitions between consecutive baskets, it establishes a transitional graph based on these relations and employs a shared self-attention mechanism to update each item's node embedding through weighted summation of neighboring nodes. This architectural design enables TransGAT to excel at capturing and learning the nuances of item transitions, ensuring more effective representation learning of transitional item relations for next-basket recommendation.

- **Basket Representation through Average Pooling:** TransGAT introduces an efficient basket representation strategy using average pooling to generate unified embeddings for baskets. This approach aggregates individual item embeddings within each basket into a single, comprehensive representation while preserving the essential characteristics of the basket's composition. The unified basket embeddings facilitate effective modeling of basket-level transitions and enable the model to capture higher-level patterns in sequential basket recommendations.

- **Empirical Validation and Insightful Analyses:** Rigorous testing of TransGAT on three distinct e-commerce datasets from TaFeng, T-Mall, and Amazon demonstrates its superior performance. Outperforming state-of-the-art methods in Hit Ratios (HRs) and Normalized Discounted Cumulative Gains (NDCGs) at cutoffs of 10 and 20, the model's efficacy is well-established. Furthermore, an ablation study and several analytical evaluations were conducted to uncover the underlying reasons for the model's effectiveness, offering valuable insights into its operational mechanisms.

1.3.4 Multi-Scale Quasi-Transformer (MSQT)

The Multi-Scale Quasi-Transformer (MSQT) addresses the first and third aspects of the **central research question** by developing neural inductive biases for multi-scale sequence modeling and handling temporal dynamics in sequential recommendation. This model operates on a transformer-based encoder-decoder architecture tailored for recommending a sequence of items in one shot, addressing the research question: *How to effectively recommend a sequence of items in one shot through better modeling item sequences and time information?* Its key contributions include:

- **Integration of Multi-Scale Convolutional Networks:** MSQT innovatively incorporates multi-scale convolutional networks within the transformer architecture. This integration enables the model to effectively capture group-level interactions among items at different granularities, facilitating the recognition of both local and global patterns in user-item interaction sequences. The multi-scale approach allows for more comprehensive sequence modeling by analyzing interactions at varying temporal resolutions simultaneously.

- **Dynamic Time Warping Embedding (DTWE):** A novel embedding technique, DTWE, has been introduced to infuse the model with temporal dynamics. This method goes beyond capturing mere timestamps; it adeptly encapsulates the relative temporal information within sequences. Such an enriched temporal understanding allows MSQT to offer a more dynamic and nuanced perspective in sequential recommendations.

- **Empirical Validation:** Rigorous testing on four real-world datasets from Amazon Reviews has positioned MSQT at the forefront of performance, outstripping contemporary methods in terms of Hit Ratios and Normalized Discounted Cumulative Gains (NDCGs) at cutoffs of 10 and 20. Additionally, an ablation study and various analyses have been conducted to explore the underlying mechanisms contributing to the model's effectiveness and to substantiate the impact of its novel features.

1.3.5 Temporal Sets Nested Transformer (TSNT)

TSNT addresses the second and third aspects of the **central research question** by developing neural inductive biases to capture transitional relations between sets of items and handle temporal dynamics in sequential recommendation. This model is specifically designed for sequential recommendations of temporal sets, where each user interaction set is associated with distinct timestamps. It addresses the research question: *How to effectively model the temporal sets sequence with time interval information for recommending next temporal sets?* TSNT's key contributions include:

- **Nested Transformer Architecture:** At its core, TSNT features a novel nested structure, comprising the Set Items Transformer (SIT) and the Temporal Sets Transformer (TST). SIT is responsible for encoding the nuances and interdependencies of items within individual sets, providing a detailed representation of each set's composition. Subsequently, TST takes these detailed set encodings and integrates them over time, capturing the temporal evolution of the sequence. This nested approach allows TSNT to encapsulate both the fine-grained item-level details and the broader temporal sequence dynamics.

- **Time Interval Quantizational Encoding (TIQE):** A central innovation in TSNT is the TIQE method. This approach quantizes and integrates time interval data within the transformer model, empowering TSNT with an enhanced capability to comprehend and leverage the temporal dynamics in user-item interactions. TIQE enriches the model's temporal understanding, significantly boosting its predictive accuracy.

- **Empirical Validation and Analytical Insights:** TSNT's effectiveness has been rigorously validated across several renowned e-commerce datasets, including TaFeng, T-Mall, and Android Apps from Amazon Reviews. Demonstrating superior performance, TSNT outpaces existing benchmarks in metrics such as Hit Ratios and Normalized Discounted Cumulative Gains (NDCGs) at both 10 and 20. Further, detailed ablation studies and analytical examinations provide valuable insights into the model's components, revealing the mechanisms driving its remarkable performance and the efficacy of its integrated set and time interval representations.

1.4 Thesis Outline

This thesis is systematically structured into the following chapters:

1. **Chapter 2 - Literature Review:** This chapter presents a comprehensive survey of sequential recommendation systems, examining them through the lens of inductive bias. It systematically categorizes and analyzes various forms of inductive biases in sequential recommendation, with particular emphasis on neural inductive bias, illuminating key trends and recent advances in the field.

2. **Chapter 3 - Multi-Scale Quasi-RNN (QR-Rec):** This chapter introduces QR-Rec, a novel neural architecture for next-item recommendation. It provides an in-depth examination of the model's architecture, extensive experimental validation across multiple datasets, and thorough analysis of its performance characteristics.

3. **Chapter 4 - Transitional Graph Attentional Net (TransGAT):** Dedicated to next-basket recommendation, this chapter presents TransGAT and

its innovative graph attention-based architecture. It offers comprehensive coverage of the model's theoretical foundations, experimental methodology, and detailed performance evaluation.

4. **Chapter 5 - Multi-Scale Quasi-Transformer (MSQT):** This chapter explores MSQT, an advanced encoder-decoder architecture for next-items recommendation. It thoroughly examines the model's design principles, experimental framework, and conducts rigorous analysis of its effectiveness through extensive empirical studies.

5. **Chapter 6 - Temporal Sets Nested Transformer (TSNT):** This chapter introduces TSNT, a sophisticated nested transformer architecture for temporal sets recommendation. It provides comprehensive coverage of the model's innovative design, experimental validation, and in-depth analysis of its performance across various metrics.

6. **Chapter 7 - Conclusion and Future Work:** The final chapter synthesizes the key contributions and insights from the research, while outlining promising directions for future investigation in sequential recommendation systems on the basis of the research presented in this thesis.

Chapter 2

Literature Review

2.1 Overview

Inductive bias [32] encompasses the assumptions and predispositions introduced into the learning process, guiding models to generalize from limited data to broader scenarios. From a holistic perspective, this concept extends beyond model choice and algorithm design, permeating every decision and procedure within the machine learning pipeline. This includes, but is not limited to, data preprocessing, data splitting, representation methods, objective functions, optimization strategies, evaluation metrics, and deployment practices. Effectively integrating inductive biases across all these stages is crucial for enhancing the performance and generalization capabilities of machine learning models. In the context of sequential recommendation systems, which predict user preferences and behaviors over order and time, neural inductive biases play a significant role. These biases, inherent in neural network architectures, influence how models process and learn from sequential data. Recent advancements in deep learning have underscored the importance of these biases in improving the performance of sequential recommendation systems, enabling more accurate predictions of various forms of next-entities in user interactions. This chapter provides a comprehensive review of the literature on various inductive biases across different stages of the pipeline, with a particular emphasis on neural inductive biases and their role in enhancing the effectiveness of sequential recommendation systems. It synthesizes over eight years of research, from 2016 (and before) to mid-2024, summarizing key trends and developments.

Following this statistical overview, a unified definition and formulation are introduced to integrate various types of sequential recommendation setups, which are often considered independently in existing studies. To demonstrate the relevance of this review to the works in this thesis, a dedicated subsection outlines the connections between the literature review and the specific models and methodologies discussed in subsequent chapters. To further clarify the landscape, a novel taxonomy is introduced, categorizing inductive biases across different stages of the sequential recommendation pipeline, with a particular focus on neural inductive biases as a subset of architecture inductive biases. This approach helps structure the diverse methodologies and provides a clearer understanding of their relative merits. Finally, a conclusion section finalizes this literature review by summarizing the key findings and insights gained from this comprehensive analysis.

2.2 Introduction

In the era of information abundance, recommendation systems have become essential tools for online services, helping users navigate vast amounts of data and receive personalized content [33]. The rapid growth of data and advancements in deep learning have driven the development of sophisticated recommendation models that leverage neural architectures to enhance accuracy and adaptability across various domains. Among these systems, sequential recommendation [34–38] focuses on modeling the ordered progression of user interactions. By analyzing historical sequences and potentially incorporating contextual information, sequential recommendation systems aim to predict the most relevant items for users' future interactions. This approach is particularly valuable in dynamic online environments, where understanding evolving user preferences is crucial for delivering engaging and personalized experiences.

However, capturing the complexities of dynamic user preferences introduces challenges that go beyond surface-level interactions. One significant challenge arises from the various biases inherent in both the data and the algorithms used. These biases, which include confirmation bias, sampling bias, and inductive bias, can profoundly impact the learning and generalization capabilities of recommendation systems, often making them difficult to avoid and recognize. From a more specific perspective, inductive bias [32, 39] refers to the inherent assumptions present in any

algorithm or model, influencing its configurations, architectures, mechanisms, and hyperparameters. These biases enable models to adapt effectively to the provided data, but they may also limit an algorithm's flexibility in dealing with diverse data types or challenges.

Though there are not that simple utilization of very pure inductive biases to the sequential recommendations field, simple thoughts still can be induced if one wants to think in that direction. For example, **Linear and Logistic Models** assume a linear relationship between input variables and outcomes. While simple linear models are not commonly used in the field of sequential recommendation systems, they can be beneficial for capturing straightforward and consistent trends in user behavior over time. For instance, if a user's interest in a certain genre or type of product increases or decreases steadily, linear models can effectively predict future preferences based on this trend. However, these models may struggle to accurately capture the more complex, non-linear patterns typical of user behavior, such as abrupt shifts in interests or diverse interaction habits that cannot be easily approximated by a straight line. Moving on, similar to linear models which are not solely used, **Support Vector Machines (SVMs)** [40] hinge on the assumption of linear separability in a transformed, higher-dimensional space. In the context of sequential recommendation, SVMs could be adept at identifying distinct phases or states within a user's interaction history by mapping these sequences into a space where different behavioral states are distinguishable. For example, SVMs can effectively segment sequences into different clusters based on user interactions that exhibit clearly separable patterns. However, their effectiveness diminishes when user behavior phases overlap significantly or when transitions between states are subtle and not clearly separable. Meanwhile, **Decision Trees**[41] assume that decision boundaries can be expressed through discrete feature-based splits. In sequential recommendation systems, they can capture non-linear relationships between user actions and preferences across different points in time. This is particularly useful for modeling complex decision rules, such as when a user's future interest is highly dependent on specific interactions in their past sequence. For instance, it can help model how a user's interest in particular types of content evolves based on a sequence of prior interactions. However, they can be prone to overfitting and sensitive to slight variations in sequential data, leading to unstable recommendations. Building on Decision Trees, **Random Forests**[42] and **Gradient Boosting Machines (GBMs)** [43] like XGBoost[44] and LightGBM[45] enhance prediction

accuracy by aggregating multiple trees. In sequential recommendation systems, these methods combine insights from different parts of a user's interaction history to generate robust and accurate predictions. Random Forests, by averaging the predictions from many trees, reduce the impact of outliers and noise in sequences, making them effective for diverse and variable user interactions. GBMs, on the other hand, sequentially correct prediction errors, allowing them to capture and refine evolving user preferences over time. This aggregated approach helps in providing stable and accurate recommendations for users with dynamic behavior. **k-Nearest Neighbors (k-NN)** [46] rely on the assumption that similar data points are likely in proximity. In sequential recommendation, this translates to the notion that users with similar historical interaction sequences are likely to have similar future preferences. k-NN can be effective in identifying local patterns in user sequences, recommending items based on the similarity of past interactions. For example, if a user's interaction sequence closely matches those of others who preferred certain items, k-NN would recommend those items. However, k-NN struggles with high-dimensional data and long interaction sequences, where the concept of "closeness" becomes less meaningful. **Naive Bayes classifiers**[47] assume independence among predictors. For sequential recommendation, they can quickly estimate the likelihood of future interactions based on the frequency of individual past interactions, assuming these events are independent. This approach can be useful for fast and scalable recommendations when detailed dependencies among interactions are not critical. However, the independence assumption often does not hold in practice, as user interactions are typically influenced by previous actions. As a result, Naive Bayes classifiers might miss important sequential patterns and interactions necessary for accurate recommendations. **k-Means Clustering**[48] partitions data into k distinct clusters, assuming spherical clusters of similar size. In the context of sequential recommendation systems, k-Means can group users based on the similarity of their interaction sequences, facilitating personalized recommendations for each cluster. For example, users with similar consumption patterns can be clustered to receive similar content suggestions. However, the method's effectiveness can be limited by its sensitivity to initial conditions and the assumption that clusters are spherical and equally sized, which may not reflect the diversity and complexity of real-world user interactions. **Principal Component Analysis (PCA)**[49] presupposes that variance indicates importance. In sequential recommendation, PCA can reduce the dimensionality of interaction sequences by highlighting the main

components that capture significant user behavior trends. This dimensionality reduction helps in simplifying the recommendation process by focusing on the most influential aspects of user interactions. For instance, PCA can identify dominant patterns in a user's interaction history, aiding in more effective recommendation. However, PCA might overlook complex, non-linear relationships within sequences that are crucial for a detailed understanding of user preferences.

As machine learning progresses, the focus on inductive biases extends to neural networks, leading to the concept of **Neural Inductive Bias**, which is the main focus of this chapter. This includes assumptions or prior knowledge embedded within neural network architectures and configurations, significantly influencing how these networks learn and generalize in sequential recommendation systems. **Multi-Layer Perceptrons (MLPs)** [50] assume that complex functions can be approximated through layers of linear mappings and non-linear activations. In sequential recommendation, MLPs can model the relationships between sequences of user interactions and their future preferences by processing historical data through multiple layers. This allows MLPs to learn complex patterns across sequences and adapt to various types of sequential data, such as identifying how a user's interest in specific content types evolves over time. However, MLPs might be inefficient for capturing long-term dependencies without additional mechanisms to handle temporal dynamics in the sequences. **Convolutional Neural Networks (CNNs)** [51] leverage the assumption that local spatial correlations and hierarchical features in input data can be effectively captured with fewer parameters. In the realm of sequential recommendation, CNNs can capture local dependencies within a sequence of user interactions. For example, CNNs can model how recent interactions influence immediate future preferences by applying convolutional filters over the sequence. This makes them well-suited for capturing short-term behavior patterns. However, CNNs may struggle with long-range dependencies across sequences unless augmented with specialized techniques like dilated convolutions. **Recurrent Neural Networks (RNNs)** [52] are designed with the bias that data exhibits temporal or sequential dependencies. RNNs are particularly effective in sequential recommendation systems for modeling how past user interactions influence future preferences over time. By maintaining a hidden state that evolves with each new interaction, RNNs can capture the temporal dynamics of user behavior. This allows them to provide personalized recommendations based on the sequence of a user's interactions, such as predicting the next product a user might be interested

in based on their purchase history. However, RNNs can encounter difficulties with long sequences due to issues like vanishing gradients, which can hinder their ability to capture long-term dependencies. The **Transformer** architecture [30] revolutionized natural language processing with its assumption that understanding data requires capturing global dependencies. In sequential recommendation, Transformers use self-attention mechanisms to effectively manage long-range relationships in user interaction sequences. This allows them to model how earlier interactions in a sequence influence current and future preferences, regardless of their position in the sequence. Transformers are particularly powerful in scenarios where understanding the context of interactions over long sequences is crucial, such as recommending the next article or video based on a user's extensive browsing history. Their ability to capture global dependencies makes them highly effective for handling complex and diverse sequential patterns. **Graph Neural Networks (GNNs)** [53] assume that data can be structured as graphs, capturing complex relational and structural information. In sequential recommendation systems, GNNs can model the relationships between items and users as a graph, capturing how user preferences evolve based on interactions with various items. For instance, GNNs can effectively represent and leverage the connections between users and items in a recommendation scenario, such as understanding how a user's interest in movies evolves based on their interactions with related genres or directors. GNNs can capture the complex and interdependent nature of user-item interactions, providing a powerful tool for modeling and predicting user preferences in dynamic and interconnected environments. **Mamba** [54] focuses on processing long sequences with Selective State Space Models (S4), emphasizing significant information for efficient sequence modeling. In sequential recommendation, Mamba can handle very long user interaction sequences by focusing on the most relevant parts of the sequence. This approach is particularly useful for maintaining the performance and relevance of recommendations for users with extensive and complex interaction histories. By selectively emphasizing important interactions, Mamba can efficiently capture and model the critical aspects of user behavior over long periods, enhancing the accuracy of sequential recommendations. **Diffusion Models** [55] assume data can be represented as a diffusion process, excelling in modeling high-dimensional distributions. In sequential recommendation, they can model the continuous evolution of user preferences as a diffusion process, capturing subtle changes in user behavior

over time. This approach can be particularly valuable for predicting how user interests spread and evolve across a series of interactions, such as modeling how a user's interest in different genres gradually shifts as they explore new content over time.

Each of these models embodies unique inductive biases that guide their learning and generalization processes. Understanding and leveraging these biases is essential for enhancing the performance of sequential recommendation systems. Recognizing the importance of these biases, this chapter will primarily explore neural inductive biases within sequential recommendation systems, focusing on their role as a subset of architectural inductive biases. Additionally, it will examine how other types of inductive biases influence different stages of the sequential recommendation pipeline. The aim of this literature review is to provide a thorough understanding of how inductive biases across various stages, especially neural inductive biases, contribute to the design and effectiveness of sequential recommendation systems, enhancing their ability to predict and recommend effectively.

The structure of the following sections is as follows:

- **Section 2.3**: This section provides a comprehensive statistical review of existing works in the field of sequential recommendation systems. It includes an analysis of records and trends in publications from top-tier conferences and journals, a review of existing surveys, and introduces unified definitions and formulations for various types of sequential recommendation setups. The section concludes by linking the literature review to the subsequent chapters of the thesis, highlighting connections between the review and the proposed research framework.
  - **Subsection 2.3.1**: Provides a summary of existing works in sequential recommendation from top-tier conferences and journals, offering an overview of significant advancements in the field.
  - **Subsection 2.3.2**: Reviews prior surveys on sequential recommendation and deep learning, emphasizing the unique insights each survey brings to the field.
  - **Subsection 2.3.3**: Proposes a unified definition and formulation for different types of sequential recommendation setups, emphasizing core and

side inputs, outputs, and the importance of maintaining the sequence order.

  - **Subsection 2.3.4**: Discusses the relevance of the literature review to the subsequent chapters of the thesis, linking the review to the later proposed research framework.

- **Section 2.4**: Introduces a novel taxonomy categorizing inductive biases across various stages of the sequential recommendation pipeline. This taxonomy focuses on the different types of inductive biases and their roles in shaping the performance of sequential recommendation systems.

  - **Subsection 2.4.1**: Covers the data preprocessing inductive bias and its effects on sequential recommendation.
  - **Subsection 2.4.2**: Looks at the data split inductive bias and its role in sequential recommendation.
  - **Subsection 2.4.3**: Explores the representation inductive bias and its influence on how data is represented and utilized in sequential recommendation.
  - **Subsection 2.4.4**: Examines the architecture inductive bias, especially the neural inductive biases, discussing how different neural network architectures impact the performance of sequential recommendation systems.
  - **Subsection 2.4.5**: Delves into the objective inductive bias, focusing on the choice of objective functions and their implications for system performance.
  - **Subsection 2.4.6**: Analyzes the optimization inductive bias, looking at how optimization strategies affect the learning process in sequential recommendation systems.
  - **Subsection 2.4.7**: Reviews the evaluation inductive bias within the context of sequential recommendation.
  - **Subsection 2.4.8**: Discusses the deployment inductive bias and how it affects sequential recommendation systems.

- **Section 2.5**: Concludes the chapter by summarizing the contributions and insights from the literature review.

2.3 Review of Existing Works

This section provides a comprehensive review of existing works in the field of sequential recommendation systems, focusing on seminal contributions from top-tier conferences and journals. The review covers more than eight years of research conducted from 2016 (and before) to mid-2024, highlighting key works and trends in the domain. The goal is to offer a detailed overview of the evolution of sequential recommendation systems and the diverse inductive biases employed to enhance their performance. The section also reviews existing surveys on sequential recommendation and deep learning, providing insights into the unique perspectives and contributions of each work. Furthermore, it proposes a unified definition and formulation for different types of sequential recommendation setups, aiming to streamline the understanding and analysis of sequential recommendation systems. Finally, it categorizes the works based on different sequential recommendation problems, such as next item(s) recommendation, next basket recommendation, and temporal sets recommendation, introduces and explains their application scenarios, and presents their relatedness to the later chapters.

2.3.1 Summary Statistics

This subsection provides an in-depth review of the major advancements in sequential recommendation systems up until mid-2024, focusing on seminal works published over the past eight years in top-tier conferences and journals related to recommendation systems, data mining, machine learning, and artificial intelligence. Due to the ongoing publication cycle, only a partial record for 2024 is available, including some work names without citations. The review highlights contributions from conferences such as AAAI, CIKM, ICDE, ICDM, ICLR, ICML, IJCAI, KDD, NEURIPS, RECSYS, SIGIR, WSDM, and WWW, and from journals such as TKDE and TOIS. The collection of works was conducted by searching keywords including, but not limited to, 'sequential recommendation', 'session-based recommendation', 'next item recommendation', 'next basket recommendation', and 'temporal sets recommendation' in the proceedings of each year of the conferences and journals as well as on platforms like Google Scholar. Some works that do not mention the keywords in the title but are related to the sequential recommendation field were also included. Despite extensive efforts to ensure the completeness of the

review, it is possible that some works may have been inadvertently omitted. If any such works are found in the future, they will be added to the content.

In recent years, there has been a substantial increase in research output in the area of sequential recommendation systems, evidenced by the growing number of publications. To illustrate this trend, around 530 pertinent works have been recorded in the tables. These works categorize seminal contributions in the domain, tracing the evolution of sequential recommendation systems from their early stages prior to 2016 through to the latest advancements in 2024. Each listed work represents a pivotal contribution to the field, highlighting various methodological breakthroughs, ranging from initial sequential models to more contemporary approaches, including but not limited to attention-based models, graph neural networks, and self-supervised learning techniques. The tables underscore the diverse array of conferences and journals that have contributed to the field, reflecting the interdisciplinary nature of sequential recommendation research. This comprehensive overview provides a valuable resource for understanding the progression of sequential recommendation systems and the various inductive biases that have been employed to enhance their performance.

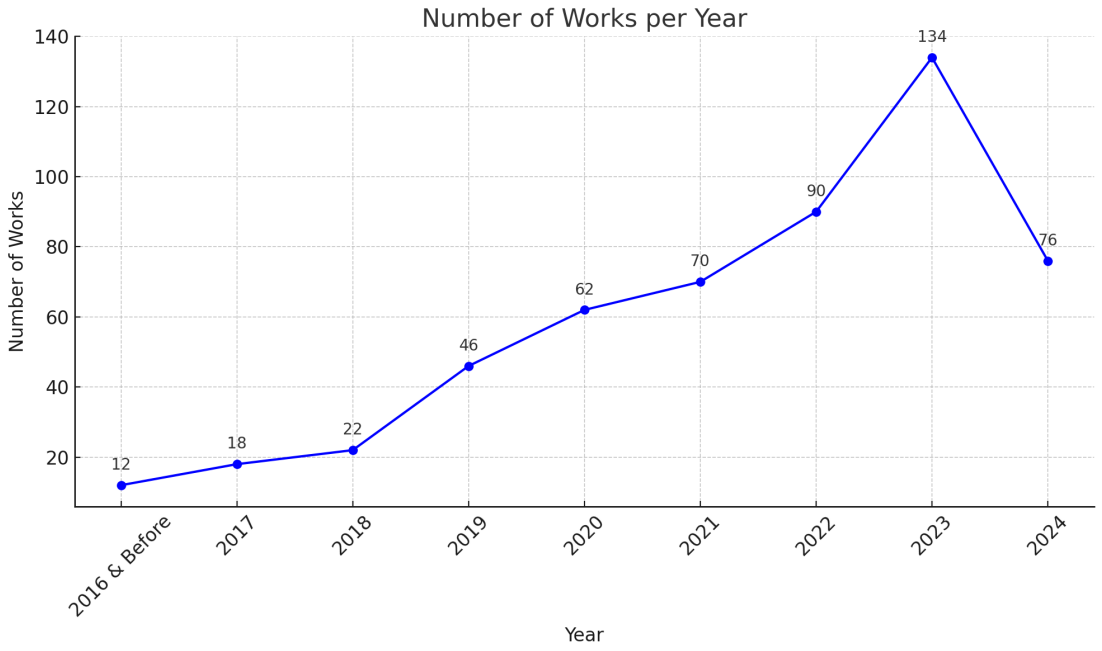


FIGURE 2.1: No. of Papers v.s. Year

Additionally, a graphical representation of the number of papers published annually is provided in Figure 2.1, offering a visual perspective on the expanding research

TABLE 2.1: Works Until 2020

C/J	2016 & Before	2017	2018	2019	2020
AAAI			ATEM[56] ATRank[57] AttRec[17]	SRGNN[21] RepeatNet[58] MARank[59] HCRNN[60]	RKSA[19] MAGNN[23]
CIKM		NARM[61] MrRNN[62] PGE[63] ListNetSIE[64]	GRU4Rec+[65] VRM[66] triple2vec-adaLoyal[67]	history-tell[68] BERT4Rec[69] SelCa[70] CosRec[12] HMN[71] DCNSR[72] FGNN[22]	DynamicRec[73] SGNNHN[74] MTAM[75] S3Rec[76] IDSR[77]
ICDE				AIR[78]	
ICDM	Fossil[2] CARNN[79]		SASRec[16]	LINet[80]	CITIES[81] DGTN[82]
ICLR	GRU4Rec[6]				
IJCAI		TimeLSTM[83] SPMC[4] BFM/CBFM[84] SWIWO[85]	RCF[86] SHAN[87] TransRec[88] BSEQ[89]	DIBPEB[90] SDIV[91] RNS[92] MCPRN[93] GCSAN[94] FDSA[18] ISLF[95] Beacon[96]	CoSAN[97] MAN[98]
KDD	basket random walk[99]	CVAE[100] RPF[101]	STAMP[102] BINN[103]	DEEMS[5] HGN[104] SSRM[105] Sets2Sets[28]	DSSRec[106] ComiRec[107] grad-align-optim[108] GeoSAN[109] seq2seq-self-supervise-distangle-rec[106] LESSR[110]
RECSYS	NNRec[27] GRU4Rec++[111] ALMM[112] NN-BPR-IE[113]	CRNNs[114] ReLaVaR[115] EDRec[116] HRNN[9] 3DCNN[13] User based GRU[117]	TransFM[118] action-RNN[119]	pred-limit[120] CDB[121]	CAGE[122] SSE-PT[123] home-prod[124] CoSeRNN[125] FISSA[126] MEANTIME[127] longitudinalEffect-session[128]
SIGIR	DREAM[7] HRM[26]		KSR[129] KrNN[130] ANAM[29]	CTRec[131] CSRM[132] π-Net[133] STAN[134] GSU[135] HPMN[136]	GLSGRL[137] HyperRec[138] MKMSR[139] Chorus[140] KERL[141] MFGAN[142] CpRec[143] MRIF[144] SRIEM[145] ICMSR[146] TAGNN[24] TASER[147] HLN[148] SAO[149] GCE-GNN[25] GAG[150] SGS[151] PeterRec[152] DSNTSP[31] NARE[153] TIFUKNN[154]
WSDM		RRN[8] NSR[155]	RUM[156] CASER[11]	DGRec[71] NextItNet[14] TMRN[157] THRNN[10] SVAE[158]	CAR[159] TiSASRec[20] LARA[160] OAR[161] KAMemNN[162] SCoRe[163]
WWW	FPMC[1] HCM[3]			HierTCN[15] STAR[164] HNVM[165] RCNN[166] VASER[167] SLRC[93]	IMfOU[168] CTA[169] GRec[170] ESRM-KG[171] ASLI[172] HierTrans[173] LOKI[174]
TKDE			MV-RNN[175] RPF[176]	CTransRec[177]	A-PGNN[178] H4MF[179]
TOIS				BIS[180]	KDA[181] FGNN[182]

TABLE 2.2: Works from 2021 to 2022

C/J	2021	2022
AAAI	DHCN[183] MTD[184] SkipRec[185] DSAN[186] NOVA[187] DMAN[188] Mecos[189]	FPAdaMetric[190] DiPs[191] Gre4RecBE[192]
CIKM	CBML[193] Seq2Bubbles[194] KGIE[195] STEN[196] H2SeqRec[197] LSAN[198] Locker[199] DRL-SRe[200] COTREC[201] TGSRec[202] DT4SR[203] CauSeR[204]	DualRec[205] PERIS[206] EC4SRec[207] MCLSR[208] ContrastVAE[209] KuaiRand[210] CBiT[211] TLSRec[212] C2DSR[213] GNG-ODE[214] STAGE[215] NextIP[216] TCPSRec[217] DEPS[218] HSD[219] MAE4Rec[220] ReDA[221] MML[222] Query-SeqRec[223] ISCON[224] FwSeqBlock[225] ML-1M++[226] ACRec[227]
ICDE	VSAN[228] DHINs[229]	DCAN[230] EMBSR[231] CL4SRec[232]
ICDM	GNRRW[233] MMInfoRec[234]	CANet[235] OverRec[236]
ICLR	Openly Published but no related papers	Openly Published but no related papers
ICML	self-modulating attention[237]	Openly Published but no related papers
IJCAI	BERD[238] DA-GCN[239] RAP[240] SSI[241] PIMI[242] AOS4Rec[243]	GCL4SR[244] MLP4Rec[245]
KDD	SEMI[246]	PARSRec[247] ITPS-BERT4Rec[248] MBHT[249] DL-MIA[250] UniSRec[251]
NEURIPS	Openly Published but no related papers	CaseQ[252]
RECSYS	Transformers4Rec[253] data-free-attack[254] neural-basket-emb[255] sampling-strategy-eval[256] INSERT[257]	Rec-denoiser[258] hyper-conv-NBRR[259] GPG4HSR[260] ARD[261] Substitution-Defense[262] BERT4Rec-review[263] recency sampling[264] CARCA[265] P5[266]
SIGIR	TASRec[267] E-BART4Rec[268] ICAI-SR[269] ProxySR[270] DAT-MDI[271] CauseRec[272] MetaTL[273] MoSeR[274] CASR[275] CoCoRec[276] LightSANs[277] SURGE[278] ASReP[279] StackRec[280] CLEA[281]	MARIS[282] TSCR[283] IPSRec[284] DCN[285] PAUP[286] ReCANet[287] RESETBERT4Rec[288] Ada-Ranker[289] PFRec[290] MGNM[291] DPP[292] DIF-SR[293] session-info-bert[294] ELECRec[295] CAFE[296] news-seqrec[297] MB-STR[298]
WSDM	SINE[299] SEFrame[300] dyna-user-intent[301]	S-Walk[302] IFR[303] RecGURU[304] Trans2D[305] DuoRec[306] HG-GNN[307] Fair-SRS[308]
WWW	SLIST[309] DeepRec[310] LISA[311] ACVAE[312] RetaGNN[313]	NSCDB[314] FMLP-Rec[315] CLSR[316] STARec[317] ICLRec[318] STOSA[319]
TKDE	HAM[320] ISRec[321] graph-emb-smooth[322] LP-MRGNN[323] PSJNet[324]	RL-ISN[325] Disen-GNN[326] TGT[327] DGSR[328] S2PNM[329] CmnRec[330] LOKI[331] GIUA-GNN[332] CASR[333] M2[334]
TOIS	HAEM[335] IARS[336] MDSR[337] SCG-SPRe[338] PosRec[339] KA-MemNN[340] GCARM[341]	CGL[342] metaCSR[343]

landscape. This figure highlights a steady increase in the number of publications over the years, reflecting the growing interest and investment in sequential recommendation research. Specifically, the data shows a consistent upward trend, peaking in 2023 with 134 works. Due to the partial available data for 2024, the count is at 76, but it is expected to match or surpass the 2023 number as major venues like CIKM, ICDM, KDD, and RECSYS have not yet published their 2024 proceedings. The category "2016 & Before" includes 12 works, indicating the early contributions to the field before the rapid growth observed in subsequent years. This surge underscores the importance of understanding the evolution of sequential recommendation systems and the inductive biases that have shaped their development. The visual trend signifies the increasing recognition of the significance of sequential recommendation systems in the broader context of artificial intelligence and machine learning.

To be more specific for each venue, another histogram based subplots have been

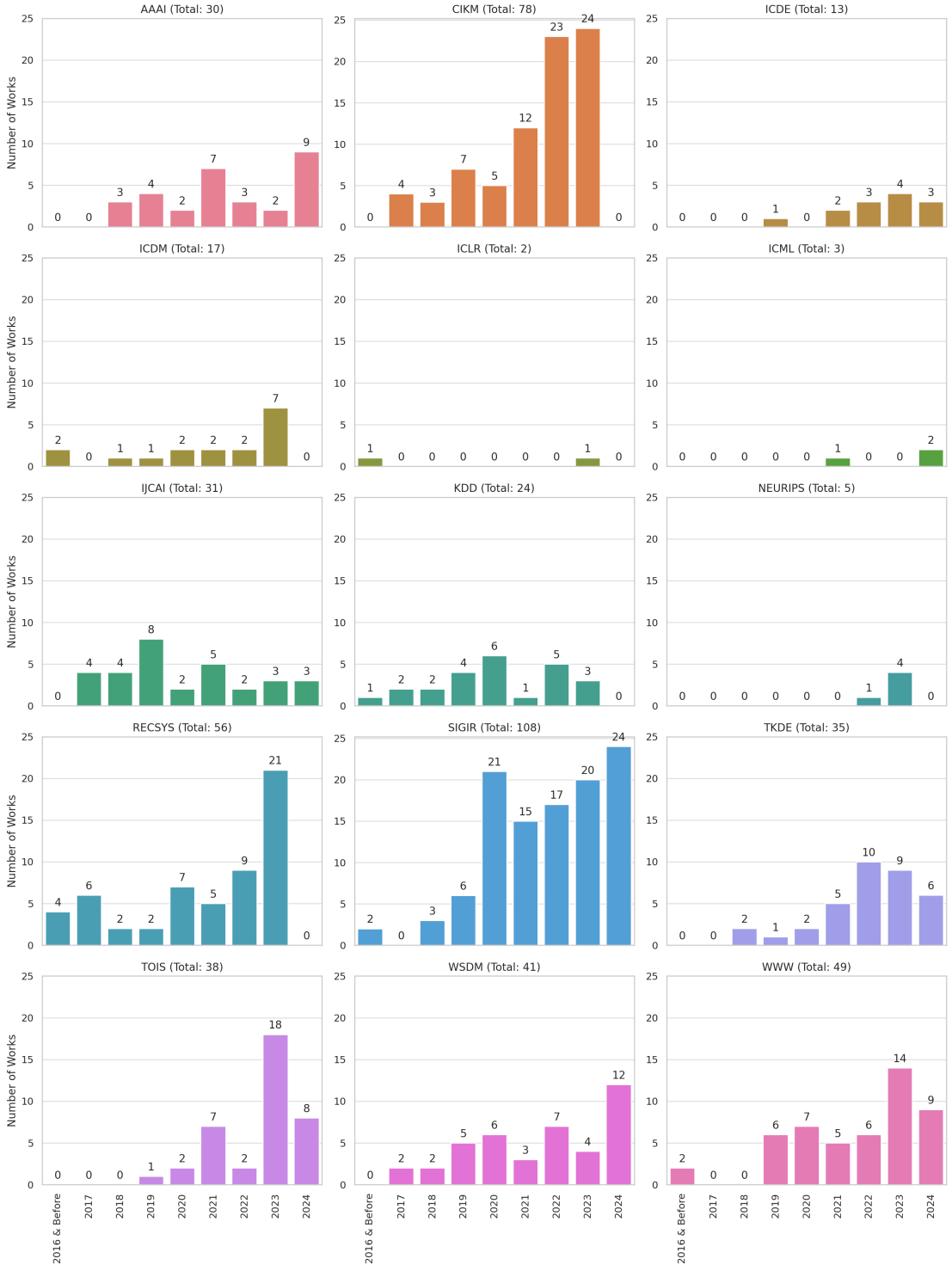


FIGURE 2.2: No. of Papers v.s. Year for Selected Top-tier Conferences and Journals

TABLE 2.3: Works from 2023 to 2024

C/J	2023	2024
AAAI	DyMuS[344] TiCoSeRec[345]	Ada-Retrieval[346] sequential-indexing[347] PDRec[348] SComGNN[349] BSARec[350] dual-view-whitening[351] SparseEnNet[352] TGCL[353] CoreRec[354]
CIKM	CPMR[355] Multiple-KVs[356] M3SRec[357] TRUF[358] APGL4SR[359] ARISEN[360] EASE[361] CDSR[362] AdaMCT[363] AutoSeqRec[364] TASTE[365] MMSR[366] RUEL[367] Prod2Vec-Var[368] SecGT[369] MSGAT[370] ODUpdate[371] DiffuASR[372] CGSR[373] TSRec[374] AC-TSR[375] ADRL[376] CCBT[377] INSPIRE[378]	Not Openly Publish anything, Conf Oct 2024
ICDE	Causer[379] IMSR[380] REKS[381] SLIME4Rec[382]	SSDRec[383] Meta-SGCL[384] MISSL[385]
ICDM	FM-IGNN[386] CCA[387] SCRIPT[388] PriCDSR[389] SIR-GNN[390] GeoMixer[391] GOSR[392]	Not Openly Publish anything, Conf Dec 2024
ICLR	ResAct[393]	Openly Published but no related papers
ICML	Openly Published but no related papers	HSTU[394] SQN[395]
IJCAI	SR-PLR[396] BRL[397] PMAN[398]	CussCtpm[399] SeeDRec STP
KDD	CT4Rec[400] ADT[401] RecFormer[402]	Not Openly Publish anything
NEURIPS	unif-normalized-embs[403] BirDRec[404] DreamRec[405] FAPAT[406]	Not Openly Publish anything
RECSYS	STDP[407] SAMRec[408] GenRec[409] incorp-time-seq[410] inte-ACT-R[411] MCV-SBR[412] SlateQ[413] TAIW[414] SEQUER[415] MGCL[416] CFIT4SRec[417] DLFS-Rec[418] DREAM[419] ECL-SR[420] gSASRec[421] BTBR[422] ReSeq[423] SPARE[424] STRec[425] SINE[426] CRR-Transformer[427]	Not Openly Publish anything, Conf Oct 2024
SIGIR	RL4RS[428] Poisoning-SSL-based-RS[429] MMNR[430] KMVG[431] MELT[432] RSR[433] LOAM[434] DigBot[435] EMKD[436] LinRec[437] MAERec[438] MCLRec[439] HPM[440] DROS[441] FEARec[442] MTAW[443] GNNO[444] MOJITO[445] APC[446] Adjusted-RR[447]	TREx[448] FineRec[449] DT4IER[450] IDGen[451] MiaSRec[452] LoRec[453] DIMO[454] LRD[455] CMCLRec[456] LLM-BRec[457] ExUA[458] FRec[459] IISAN[460] SelfGNN[461] A-FSR[462] LlaRA[463] MBASR[464] SeRALM[465] pacer-runner[466] srt[467] SCM4SR[468] PO4ISR[469] MDSR-DSL[470] SCEMIM[471]
WSDM	IOCRec[472] IDNP[473] DGNN[474] Atten-Mixer[475]	TKG-SRec[476] LRURec[477] ICSRec[478] MQSA-TED[479] MAN[480] RecJPQ[481] SRU[482] COMED[483] softmax-CPR[484] MSSR[485] DebiasedSR-DRO[486] GHGTID[487]
WWW	STEAM[488] DFAR[489] COCO-SBRS[490] MStein[491] HADCG[492] IESRec[493] GSNIR[494] PRecQ[495] DCRec[496] MMMLP[497] VQ-Rec[498] AutoMLP[499] ME-MIA[500] SDIL[501]	IntellectReq[502] END4Rec[503] DA4Rec[504] SLIM[505] AMID[506] LLM-TRSR[507] UnifiedSSR[508] Rella[509] S4Rec[510]
TKDE	FDSA-CL[511] FCLRec[512] KGDPL[513] RNMSR[514] HapCL[515] DCL[516] MMSBR[517] TiCoSeRec[518] TMER-RL[519]	IMA-EMA[520] ISTN[521] GDERec[522] PPR[523] dynamic-item-tendency-bias[524] AutoSR[525]
TOIS	DiffuRec[526] NBR-reality-check[527] MVS[528] EODRec[529] DUVRec[530] MrTransformer[531] PHGR[532] ContraRec[533] PTGCN[534] MELOD[535] CMVCDR[536] BiPNet[537] H3GNN[538] RESTC[539] BERD+[540] understand-diversity[541] MEGAN[542] FASTER[543]	M3T[544] TRIER[545] SoftCSR[546] SMLP4Rec[547] AutoDisenSeq-LLM[548] MELOD[549] TriMLP[550] multi-view-gnn-transformer[551]

made. The histogram plots in Figure 2.2 depict the number of works published annually in various conferences and journals from 2016 (and before) to 2024. The conferences (AAAI, CIKM, ICDE, ICDM, ICLR, ICML, IJCAI, KDD, NEURIPS, RECSYS, SIGIR, WSDM, and WWW) and journals' records (TKDE and TOIS) are used for analysis. Notably, the proceedings for 2024 have not yet been published for CIKM, ICDM, KDD, RECSYS, ICLR, and NEURIPS.

These observations lead to several key insights about the field:

- **Growth in Research Output**: The clear upward trend in the number of publications indicates a growing interest and advancements in the field of sequential recommendation systems.
- **Dominant Conferences and Journals**: SIGIR stands out as the most prolific venue, followed by CIKM, RECSYS, WWW, WSDM, IJCAI and AAAI. This suggests that these conferences are the preferred platforms for researchers to publish their work in this domain.
- **Variability in Publications**: Some venues show variability in the number of publications year-to-year, which could be due to varying themes, special issues, or changes in conference focus.
- **Pending Data Impact**: The pending data for 2024 for several conferences and journals might influence these trends, but the overall growth trajectory is expected to continue.

From a detailed perspective, several key trends are evident across various conferences and journals: AAAI experienced varied publications, peaking in 2020 and 2024, indicating a recent surge in interest. CIKM showed a consistent increase, peaking in 2023, with 2024 data pending. ICDE had a steady but low publication count, with a slight recent increase. ICDM displayed a gradual increase, also peaking in 2023, with 2024 data pending. ICLR and ICML had minimal publications, with ICLR's only in 2017 and 2024, and ICML showing a small increase in 2023 and 2024. IJCAI exhibited a fluctuating trend, peaking in 2019 and 2023, while KDD showed variable publications, peaking in 2019 and 2023, with 2024 data pending. NEURIPS had minimal publications with a slight increase in 2023, pending 2024 data. RECSYS showed a general upward trend, peaking in 2023, pending 2024 data. SIGIR saw a significant increase, peaking in 2023, while WSDM's trend fluctuated, reaching its highest in 2024. WWW demonstrated a clear upward trend, peaking in 2023. Lastly, TKDE and TOIS had steady increases, with notable peaks in 2021 and 2023, respectively.

The observations reveal distinct publication trends across different conferences and journals, with a general increase in the number of works published in recent years, particularly in 2023. However, the pending data for 2024 for several conferences such as CIKM, ICDM, KDD, RECSYS, ICLR, and NEURIPS might further influence these trends, high chance to match or surpass the 2023 numbers. These trends

underscore the growing interest and investment in sequential recommendation research and the need for continued exploration and innovation in this domain.

2.3.2 Review of Existing Surveys

The landscape of existing surveys on sequential recommendation systems and deep learning methodologies encompasses a diverse range of focal points. Notably, the surveys by [33–38, 552–555] each offer distinct insights and valuable perspectives within this domain. For instance, [552] provides a comprehensive overview of cross-domain sequential recommendation systems, focusing on the fusion of user interactions across multiple domains to enhance recommendation accuracy. Similarly, [38] systematically reviews deep learning techniques employed in sequential recommendation systems, emphasizing advancements in neural network architectures and their application in capturing user preferences over time. Meanwhile, [36, 554, 555] explore the evolution of sequential recommendation algorithms, documenting the transition from traditional collaborative filtering methods to state-of-the-art deep learning models. They discuss various modeling techniques, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and attention mechanisms, which have significantly improved recommendation performance. In another detailed analysis, [37] delves into the integration of deep learning and sequential recommendation, providing insights into how deep learning models such as transformers and graph neural networks (GNNs) have been utilized to enhance the understanding of user-item interactions. Complementing this, [35] focuses on session-based recommendation systems, particularly highlighting the use of RNNs and their variants in modeling short-term user preferences within individual sessions. Moreover, [34] examines sequence-aware recommendation systems, categorizing the approaches based on the type of sequence modeling techniques employed, such as Markov chains, factorization models, and neural networks. Additionally, [33] provides an extensive review of deep learning-based recommendation systems, discussing various neural architectures and their effectiveness in capturing complex user behaviors. Lastly, [553] offers a comprehensive study on the performance of transformers, nearest neighbors, and sampled metrics in sequential recommendation systems.

Based on the existing surveys, this literature review distinguishes itself by offering a comprehensive and up-to-date analysis of the latest research trends, methodologies, and applications in sequential recommendation systems from the perspective of inductive biases across various stages of the recommendation pipeline. This review focuses on the neural inductive biases that have shaped the field. By synthesizing the latest research findings and identifying key challenges and opportunities in the domain, this review aims to provide a holistic understanding of the sequential recommendation landscape and offer valuable insights for future research directions.

2.3.3 Unified Definitions, Notations and Formulations

Considering the diverse types of sequential recommendation found in the literature, it is essential to establish a comprehensive and inclusive unification from all. In this section, the unified definitions, notations, and formulations for sequential recommendation are presented, which are used throughout the rest of the literature reivew.

Core Input Form: An **interaction** sequence is considered as the core input for sequential recommendation task. Each interaction within the sequence is either ordered or timestamped. Interactions can be represented as a single item, a basket (set) of unordered items, or an entity consisting of multiple ordered or timestamped items (the most complex case). Various interaction types exist, as detailed in [37], including **search**, **click**, **add-to-cart**, **buy**, **share**, and more. However, in most datasets, **buy** is the only interaction type considered. In this literature reivew, **buy** is treated as the default interaction type and only mention other interaction types when they differ from **buy**.

Side Input Form: Side input includes various forms of additional information that can enhance the performance of sequential recommendation. This encompasses user profiles (e.g., demographics, preferences), item profiles (e.g., features, descriptions, categories), contextual information (e.g., time, location, device), and other domain-specific side information. Such additional data can be utilized to capture the underlying structure and dependencies in the sequential recommendation task and help improve the overall effectiveness of the model.

Output Form: The primary output form is the prediction of what the next single item will be. However, depending on real-world requirements and variations, the output can be adapted to different forms. It may include a sequence of continuous items, either with order (e.g., list, next session, or next sequence) or without order (e.g., basket or set). Alternatively, the output may comprise a sequence of discontinuous items, accounting for skipped elements, session-based recommendations, basket completion, or set completion tasks. This flexibility in the output form allows sequential recommendation models to cater to a diverse range of application scenarios.

Significance: Maintaining the order of input sequences in sequential recommendation is of paramount importance, as it can unveil a variety of intrinsic relationships that could impact prediction outcomes. These relationships may include causal, temporal, or other types of associations that could be either essential or non-essential for accurate prediction. By preserving and analyzing the order of input sequences, sequential recommendation models can better understand and capture the underlying patterns in the data, ultimately leading to more effective and accurate recommendations for users.

Given the core input form, side input form, and output form, the sequential recommendation problem can be defined as follows:

Let $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be the set of users, $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be the set of items, and $T = \{t_1, t_2, \ldots, t_l\}$ be the set of possible timestamps or ordinal positions.

The core input form consists of interaction sequences for each user $u \in U$. Let $S_u = \{(i_1, t_1), (i_2, t_2), \ldots, (i_k, t_k)\}$ represent the interaction sequence for user u, where $(i_j, t_j) \in I \times T$ indicates the interaction with item i_j at timestamp or ordinal position t_j.

The side input form consists of user profiles, item profiles, and other additional features (optional). Let P_u be the user profile vector for user u, P_i be the item profile vector for item i, and $F_{(i_j,t_j)}$ be the feature vector for the interaction (i_j, t_j).

The objective of the sequential recommendation task is to forecast the next item(s) that a user is likely to interact with, based on their historical sequence of interactions and any available side information. Let O_u denote the set of predicted items

for user u. The nature of O_u can vary based on the problem context; it could be a single item, an unordered set of items, or an ordered list of items.

The objective of sequential recommendation is to learn a function f that maps the interaction sequence S_u, user profiles P_u, item profiles P_i, and additional features F (if available) to the output O_u, The function f can be parameterized by a set of model parameters θ that are learned during the training process:

$$R_u = \arg\max_n f(S_u, P_u, P_i, F; \theta) \tag{2.1}$$

The objective is to optimize the model parameters θ to ensure that the function f accurately predicts the subsequent item(s) the user will engage with.

TABLE 2.4: Notation table for sequential recommendation

Symbol	Description
U	Set of users
I	Set of items
u	A specific user
i	A specific item
S_u	Sequence of items interacted by user u
s_t	Item at position t in the sequence
T	Length of the sequence
x_t	Feature vector for item s_t
y_t	Target label for item s_t
$f(\cdot)$	Sequential recommendation model
θ	Model parameters
$L(\cdot)$	Loss function

Unified Concepts and Notations: Sequential recommendation systems epitomize a refined segment of recommendation technologies, distinguished by their focus on leveraging the chronological sequence and timing of user interactions to forge personalized suggestions. As shown in Table 2.4, the notation used in these systems is crucial for understanding the underlying mechanisms and methodologies. These systems, transcending conventional recommendation frameworks, are adept at discerning intricate patterns in user behavior across different temporal contexts. They encapsulate session-based recommendations, which are predicated on the analysis of user activities within discrete sessions, and sequence-aware recommendations, which consider the entirety of a user's interaction history, irrespective of

session boundaries. Additionally, the concept extends to session-aware recommendations, where contextual insights from specific sessions enrich the recommendation process.

Whether predicting the next individual item a user might engage with or forecasting a subsequent set of items (next-basket recommendations), the essence of sequential recommendation lies in its ability to amalgamate various facets of user interaction into a cohesive understanding. This nuanced approach aims to enhance the personalization of user experiences by aligning recommendations with the unique temporal and behavioral patterns exhibited by users.

2.3.4 Relatedness to Subsequent Chapters

The primary aim of this literature review is to provide an overview of sequential recommendation systems from the perspective of inductive biases. It encompasses a wide range of seminal works, surveys, and trends, thereby offering a comprehensive understanding of the evolution and advancements in this research domain. This subsection specifically highlights the relevance of the literature review to the subsequent chapters of the thesis, focusing on the models and methodologies discussed in each chapter, as well as the motivation and significance of the review in the context of the overall thesis and an expansion of the works that support the gaps identified in Chapter 1 1.2.

The connection between this review and the following chapters is outlined below. As indicated by the gaps and research questions introduced in Chapter 1 1.2, the subsequent chapters will address these gaps and questions by proposing novel models and methodologies:

- Chapter 3 discusses QR-Rec, which addresses the challenge of capturing long-term dependencies in sequential data through the extension of Quasi-RNN with a multi-scale framework. This chapter delves into the integration of dynamic average pooling, a hierarchical summation aggregation strategy, and empirical validation, providing a detailed exploration of QR-Rec's contributions and effectiveness in next-item recommendation. This review summarizes other next-item recommendation works and neural inductive biases based on an RNN-variant backbone, helping readers better understand the

context and motivation behind the development of QR-Rec. It also highlights how most works lack consideration of variable-length utilization, as mentioned in Chapter 1 1.2, which is part of the central research question of this thesis.

- Chapter 4 covers TransGAT, a pioneering model incorporating a graph attention network to capture transitional relations of items across consecutive baskets. This chapter examines the architectural novelty of TransGAT, its empirical validation on various datasets, and the insights gained from its superior performance in next-basket recommendation. This review provides a comprehensive overview of graph neural networks (GNNs) and attention mechanisms, which are fundamental components of TransGAT, as well as summarizing next-basket recommendation works, to elucidate the significance of these inductive biases in sequential recommendation. Additionally, it addresses the need for more complex item relationship representations, as mentioned in Chapter 1 1.2, another aspect of the central research question of this thesis.

- Chapter 5 explores MSQT, a transformative model that integrates multi-scale convolutional networks within a Transformer architecture. This chapter details the introduction of dynamic time warping embedding (DTWE), empirical validation, and the significant advancements MSQT brings to the realm of sequential recommendation by recommending a sequence of items in one shot. Beyond introducing and summarizing seq2seq-based next-item recommendation works, this review also provides a discussion on the Transformer architecture and Convolutional Neural Networks (CNNs), which are essential components of MSQT. This emphasis clarifies the inductive biases that underpin the model, including the consideration of variable-length sequences and a more comprehensive handling of dynamic interactions, both of which were identified as gaps in Chapter 1 1.2.

- Chapter 6 focuses on TSNT, which innovates the field of temporal set recommendation through its nested transformer-based architecture. This chapter discusses the nested structure of SIT and TST, the introduction of time interval quantizational encoding (TIQE), and the empirical validation that underscores TSNT's effectiveness in capturing temporal dynamics in user-item interactions. Besides summarizing temporal set recommendation works, this

review also provides an overview of the Transformer architecture, a key component of TSNT, to elucidate the importance of this inductive bias in the context of temporal set recommendation. In doing so, it addresses the need for more complex item relationship representations and more comprehensive temporal dynamics, two crucial gaps highlighted in Chapter 1 1.2.

By establishing this foundation, the literature review ensures that each subsequent chapter builds on a robust understanding of neural inductive biases and their critical role in sequential recommendation systems. This structured approach not only reinforces the coherence of the thesis but also provides a comprehensive framework for exploring both current and future developments in the field.

To further illustrate the representative works related to each task, Table 2.5 outlines the partial list of works that have contributed to different sequential recommendation problems, since the field is vast and rapidly evolving, this list is not exhaustive but provides a snapshot of the diverse range of approaches and techniques that have been developed to address various tasks in sequential recommendation.

TABLE 2.5: Selected Works for Different Tasks of Sequential Recommendation

Problem	Works
Next Item Recommendation	Fossil[2], GRU4Rec[6], Caser[11], BERT4Rec[69], NextItNet[14], SR-GNN[21], HGN[104], DEERS[72], DyMuS[344], TiCoSeRec[345], CAGE[122], SSE-PT[123], home-prod[124], CoSeRNN[125], FISSA[126], MEANTIME[127], STEAM[488], DFAR[489], COCO-SBRS[490], MStein[491], HADCG[492], IESRec[493], GSNIR[494], IMA-EMA[520], ISTN[521], GDERec[522], PPR[523], FDSA-CL[511], FCLRec[512], KGDPL[513], RNMSR[514], HapCL[515], DCL[516], MMSBR[517], SR-PLR[396], PMAN[398], CussCtpm[399], MCPRN[93], GCSAN[94], FDSA[18], ISLF[95], SLIST[309], DeepRec[310], LISA[311], ACVAE[312], RetaGNN[313], NSCDB[314], FMLP-Rec[315], CLSR[316], STARec[317], ICLRec[318], STOSA[319], IntellectReq[502], END4Rec[503], DA4Rec[504], SLIM[505], AMID[506], LLM-TRSR[507], UnifiedSSR[508], Rella[509], S4Rec[510], HAM[320], ISRec[321], LP-MRGNN[323], PSJNet[324], RL-ISN[325], Disen-GNN[326], TGT[327], DGSR[328], S2PNM[329], CmnRec[330], LOKI[331], GIUA-GNN[332], CASR[333] ...
Next Basket Recommendation	FPMC[1], NN-Rec[27], HRM[26], DREAM[7], ANAM (NAM)[29], triple2vec-adaLoyal[67], SecGT[369], CCBT[377], BFM/CBFM[84], BSEQ[89], Beacon[96], BRL[397], basket random walk[99], neural-basket-emb[255], hyper-conv-NBRR[259], GenRec[409], TAIW[414], BTBR[422], TIFUKNN[154], CLEA[281], ReCANet[287], MMNR[430], DigBot[435], TREx[448], M2[334], HapCL[515], HAEM[335], NBR-reality-check[527] ...
Next Items Recommendation	DSSRec[106], GRec[170], AOS4Rec[243], E-BART4Rec[268], recency sampling[264], ContraRec[533], IOCRec[472] ...
Temporal Sets Recommendation	Sets2Sets[28], DSNTSP[31], DNNTSP[556], SFCNTSP[557], GLOIE[558], DHNTSP[559], ETGNN[560], CTTSP[561] ...

These works collectively demonstrate the advancements in various aspects of sequential recommendation systems, providing a comprehensive understanding of the field and the innovative approaches that have been developed to address specific challenges.

Beyond the works summarized above, Table 2.6 at the end of this chapter summarizes the four tasks addressed in the subsequent chapters on sequential recommendation systems. It highlights the types of inputs and outputs involved, the

proposed models, and their potential application scenarios as development motivations. For the Next-item Recommendation task, the QR-Rec model utilizes the user's last sets of items with timestamps to predict the next item. This approach is particularly useful in scenarios such as online retail platforms, where it recommends products based on past purchases, and streaming services, where it suggests the next show or movie to watch. Mobile app recommendations leverage this model to suggest apps users might be interested in based on their previous downloads, while news article suggestions and personalized marketing use it to tailor content and advertisements to user preferences. The Next-basket Recommendation task employs the TransGAT model to predict the next basket of items based on consecutive transitional relations. This model is beneficial for e-commerce platforms and grocery shopping apps that suggest groups of items often bought together. Fashion outfit recommendations use this model to suggest complete outfits, while home improvement stores and health and wellness product suggestions use it to recommend complementary products. For the Next-items Recommendation task, the MSQT model uses sequences of past item interactions to predict the next sequence of items. This approach is applicable in content recommendation systems like video platforms that suggest a sequence of videos to watch next, playlist generation services that create music playlists, book reading sequences that recommend the next set of books to read, e-learning course recommendations that guide users through a series of courses, and fitness training plans that suggest a series of workouts. Lastly, the Temporal Sets Recommendation task utilizes the TSNT model to predict the next sequence of sets of items based on sequences of item sets with timestamps. This model is useful in subscription boxes that send curated items periodically, monthly service platforms that provide regular service updates, seasonal product recommendations for items that are relevant at specific times of the year, periodic event planning for organizing recurring events, and multi-stage project management for recommending subsequent project tasks.

2.4 Taxonomy, Summarization, and Highlights

This section presents a taxonomy of sequential recommendation techniques, emphasizing inductive biases across various stages as depicted in Figure 2.3. These

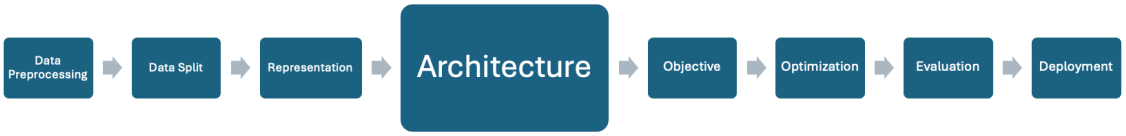


FIGURE 2.3: Sequential Recommendation Pipeline Diagram

stages include data preprocessing, splitting, representation, architecture, objectives, optimization, evaluation, and deployment, each involving various types of inductive biases that represent inherent assumptions guiding the learning process. Among these stages, the neural inductive biases within the architecture stage are the primary focus. The taxonomy classifies methods based on these biases, providing a summary and discussion of each category's key ideas. These discussions, whether or not accompanied by references, are intended to inspire design ideas for readers and the research community. Representative techniques are selected and illustrated to elucidate their principles and practical applications.

2.4.1 Data Preprocessing Inductive Biases

Data preprocessing inductive bias refers to the assumptions and predispositions made when preparing the input data for sequential recommendation tasks. These biases can affect the model's ability to learn from the data, as well as its generalization to new scenarios. Some common data preprocessing inductive biases in sequential recommendation include:

Sampling techniques are vital in sequential recommendation to manage data volume and ensure representative subsets for model training. Data sampling methods, such as uniform sampling or time-based sampling, can select subsets of the data that reflect different aspects of user interaction dynamics. For example, uniform sampling might randomly select interactions $\{(u_i, v_i, t_i)\}_{i=1}^{n}$ without considering their temporal order, potentially missing the sequential patterns crucial for recommendation. In contrast, time-based sampling selects interactions based on their timestamps $\{(u_i, v_i, t_i) \mid t_i \leq t_{\text{cutoff}}\}$, preserving temporal dynamics and helping the model learn how user preferences evolve over time. Additionally, sampling techniques are commonly employed to provide negative samples in the training data,

which is essential for training effective recommendation models. Furthermore, sampling is used to select a small, manageable corpus of testing items, significantly enhancing evaluation efficiency by reducing computational overhead while maintaining the integrity of performance assessments.

Augmentation techniques such as item shuffling, time warping, or session splitting can be employed to artificially increase the training data and improve the model's generalization. However, the choice of augmentation method may introduce bias by emphasizing certain data aspects while neglecting others. For instance, item shuffling within sessions shuffle($\{v_i\}_{i=1}^n$) helps the model learn the importance of item order but may disrupt the temporal dynamics of user preferences. Another example is time warping, where timestamps are adjusted to simulate different interaction speeds, which can help the model generalize better but might introduce unrealistic temporal patterns and noise. Session splitting, on the other hand, divides long sequences into shorter sessions to capture different user preferences but may overlook long-term dependencies in the data. By carefully selecting and balancing these augmentation methods, researchers can enhance the model's ability to learn from the data while avoiding bias.

Feature Extraction methods transform raw data into more compact and informative representations for model training. Techniques like dimensionality reduction or embedding learning prioritize certain features or relationships in the data over others. For example, using item embeddings $\mathbf{e}_v = f(v)$ learned from a pre-trained model might emphasize semantic similarity between items but not capture the unique aspects of user preferences in the target domain. In sequential recommendation, embeddings are often learned to map items into a latent space where similar items are closer together, which helps in predicting the next item in the sequence.

Data Cleaning is critical to improve the quality of the input data for model training. Techniques such as outlier detection, missing value imputation, or noise reduction can make certain assumptions about the data distribution or the nature of the noise, potentially introducing bias. For example, using mean imputation $\tilde{x}_i = \frac{1}{n}\sum_{j=1}^n x_j$ for missing values might underestimate the variability in the data, leading to a less robust model. In sequential recommendation, cleaning noisy interaction data is crucial to ensure the model learns accurate patterns of user behavior.

Normalization and Scaling are commonly used to standardize data, ensuring that features contribute equally to the model learning process. Normalizing user-item interaction data to a common scale can prevent features with larger ranges from dominating the learning process. For instance, if user interaction times are on different scales (e.g., seconds vs. days), normalizing these times to a common scale can improve the model's ability to learn temporal patterns. This process can be mathematically represented as:

$$x' = \frac{x - \mu}{\sigma}$$

where x is the original value, μ is the mean of the feature, σ is the standard deviation, and x' is the normalized value. Improper normalization, however, can introduce bias, especially if it disrupts inherent relationships within the data.

Temporal Data Smoothing involves techniques like moving averages or exponential smoothing to handle noise and fluctuations in temporal data. These methods enhance the clarity of trends in user behavior but might obscure sudden significant changes in preferences. For example, applying a moving average to a user's interaction frequency can smooth out short-term fluctuations, helping the model focus on long-term trends. The moving average can be represented as:

$$\mathrm{MA}_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i}$$

where MA_t is the moving average at time t, n is the window size, and x_{t-i} are the observed values. Exponential smoothing can be represented as:

$$\mathrm{ES}_t = \alpha x_t + (1 - \alpha)\mathrm{ES}_{t-1}$$

where ES_t is the exponentially smoothed value at time t, α is the smoothing factor, x_t is the observed value, and ES_{t-1} is the previously smoothed value. While these techniques help in focusing on long-term trends by smoothing out short-term fluctuations, they might also mask important spikes in activity that indicate a sudden interest in a new item or category, which needs to be carefully considered in the model design.

Sequence Padding and Truncation are techniques used to handle sequences of varying lengths in sequential recommendation. Padding sequences to a fixed length ensures uniformity in model input but may introduce bias by adding artificial elements. For example, padding a short sequence $\{v_1, v_2\}$ to match the length of a longer sequence $\{v_1, v_2, ..., v_n\}$ might affect the model's understanding of sequence length. Conversely, truncating sequences can lead to the loss of crucial information, especially in cases where long-term dependencies are vital for accurate recommendations. Given the intrinsic variation in the lengths of users' interaction history sequences, padding and truncation are essential techniques to ensure that the model can handle sequences of different lengths effectively without compromising the integrity of the data.

Masking Sequence is a technique used to mask certain elements in the sequence to prevent the model from learning spurious correlations. For example, in a masked language model, certain tokens are masked during training, and the model is tasked with predicting the masked tokens based on the surrounding context. Mathematically, if a sequence $S = \{s_1, s_2, \ldots, s_n\}$ contains elements to be masked, the masked sequence S' can be represented as $S' = \{s_1, s_2, \ldots, [MASK], \ldots, s_n\}$, where $[MASK]$ denotes the masked element. The objective is to minimize the loss function:

$$L = -\sum_{t \in M} \log P(s_t \mid S_{\setminus t}; \theta)$$

where M is the set of masked positions, $S_{\setminus t}$ represents the sequence with the masked positions removed, and θ denotes the model parameters. In sequential recommendation, masking can be used to simulate missing interactions or to prevent the model from overfitting to specific patterns in the data. By masking certain elements in the sequence, the model is forced to learn more robust representations that generalize better to unseen data. This approach encourages the model to focus on the underlying structure of the data rather than memorizing specific sequences, thereby improving its ability to make accurate recommendations in new contexts.

By carefully selecting and balancing these data preprocessing methods to align with the specific goals and characteristics of the sequential recommendation task, researchers can ensure that their models are better able to learn from the data and generalize to real-world scenarios.

2.4.2 Data Split Inductive Biases

The inductive bias in data splitting pertains to the presumptions made during the partitioning of data into training, validation, and testing sets for sequential recommendation. The partitioning approach can directly affect the model's learning efficacy and its generalization to new scenarios. Common biases in sequential recommendation data splitting include:

Random Split divides the data randomly into training, validation, and testing sets. This approach, while straightforward to implement, overlooks the sequential nature of recommendation tasks and may not accurately reflect a model's performance in real-world scenarios where user preferences and item popularity evolve over time. For example, if the interaction dataset is denoted by $D = \{(u_i, v_i, t_i)\}_{i=1}^{N}$, where u_i represents the user, v_i the item, and t_i the timestamp, a random split would partition D such that each (u_i, v_i, t_i) is independently assigned to training, validation, or test set. This neglects any temporal ordering.

Time-Based Split organizes the dataset chronologically, with training data composed of earlier interactions and validation and testing data comprising later interactions. It acknowledges the temporal dimension of user interactions, offering a more realistic evaluation of a model's predictive capabilities over time. For example, if D is sorted by t_i and split it such that the first 80% of interactions are used for training, the next 10% for validation, and the last 10% for testing, the temporal structure is preserved:

$$D_{\text{train}} = \{(u_i, v_i, t_i) \mid 1 \leq i \leq 0.8N\}$$

$$D_{\text{val}} = \{(u_i, v_i, t_i) \mid 0.8N < i \leq 0.9N\}$$

$$D_{\text{test}} = \{(u_i, v_i, t_i) \mid 0.9N < i \leq N\}$$

However, its effectiveness can be compromised by non-representative training data or abrupt shifts in user behavior patterns. A common strategy is leave-one-out splitting for sequential recommendation inputs data, where the last interaction of each user's interaction sequence is used for validation or testing, and the rest are used for training.

User-Based Split distributes the data according to users, ensuring that the interactions of each user are exclusively in either the training, validation, or test set. This guarantees a varied representation of users in each dataset, though it may neglect the time-sensitive aspects of user preferences and the changing popularity of items. If U denotes the set of users, this split can be defined such that $U_{\text{train}}, U_{\text{val}}, U_{\text{test}}$ are disjoint subsets of U:

$$U_{\text{train}} \cup U_{\text{val}} \cup U_{\text{test}} = U$$

$$D_{\text{train}} = \{(u_i, v_i, t_i) \mid u_i \in U_{\text{train}}\}$$

$$D_{\text{val}} = \{(u_i, v_i, t_i) \mid u_i \in U_{\text{val}}\}$$

$$D_{\text{test}} = \{(u_i, v_i, t_i) \mid u_i \in U_{\text{test}}\}$$

This split relies on the assumption that the model can generalize across different user profiles.

Session-Based Split allocates data based on user sessions, respecting the sequential order of interactions within each session. This approach is particularly suited for evaluating a model's performance on sequential recommendation tasks, as it maintains the integrity of individual user sessions. For example, if S denotes the set of sessions, the split can be defined such that $S_{\text{train}}, S_{\text{val}}, S_{\text{test}}$ are disjoint subsets of S:

$$S_{\text{train}} \cup S_{\text{val}} \cup S_{\text{test}} = S$$

$$D_{\text{train}} = \{(u_i, v_i, t_i) \mid \text{session}(u_i) \in S_{\text{train}}\}$$

$$D_{\text{val}} = \{(u_i, v_i, t_i) \mid \text{session}(u_i) \in S_{\text{val}}\}$$

$$D_{\text{test}} = \{(u_i, v_i, t_i) \mid \text{session}(u_i) \in S_{\text{test}}\}$$

However, the approach can be complicated by the absence of clear session delineations or significant variations in session lengths and the nature of interactions.

By carefully selecting data split methods that align with the specific goals and characteristics of the sequential recommendation task, researchers can ensure that their models are better able to learn from the data and generalize to real-world scenarios.

2.4.3 Representation Inductive Biases

Representation inductive biases refer to the inherent assumptions made by a technique regarding the representation of inputs, typically concerning items, sequences of items, or sets of items, based on the underlying data structure. These biases influence how the data is prepared for subsequent processing stages. In the context of sequential recommendation, the choice of representations for items, users, and supplementary information is crucial in determining the system's performance. The following categories of representation inductive biases are commonly observed in sequential recommendation techniques:

ID-based Representation employs unique identifiers, such as user or item IDs, to denote entities within the technique. Many contemporary sequential recommendation systems (e.g., [6, 11, 16]) utilize this method due to its simplicity and computational efficiency. Despite its straightforward nature, ID-based representation lacks the capacity to encode semantic information, which limits the model's ability to discern complex relationships. Standard techniques include one-hot or multi-hot encodings, transforming IDs into sparse, binary vectors. For instance, a one-hot encoding for item i can be represented as:

$$\mathbf{v}_i = [0, 0, \ldots, 1, \ldots, 0] \tag{2.2}$$

where only the i-th position is 1 and all others are 0. In sequential recommendation, this representation can lead to sparse interaction matrices, potentially failing to capture the temporal dynamics of user interactions effectively. Although later they are generally fed into embedding layers to learn more expressive representations, the initial ID-based representation can restrict the model's ability to capture nuanced relationships between items and users.

Non-ID-based Representation leverages more complex and semantically rich information through embeddings. Unlike ID-based methods, embeddings provide dense, continuous vector representations that encapsulate the intrinsic attributes of items or users, enabling more nuanced similarity measures. This enhances the personalization and accuracy of recommendations. An embedding vector for item i can be expressed as:

$$\mathbf{e}_i = [e_{i1}, e_{i2}, \ldots, e_{id}] \tag{2.3}$$

where d is the dimensionality of the embedding space. In sequential recommendation, embeddings can adapt to the evolving preferences of users by updating as new interactions occur. ATEM [56] is an example of a non-ID-based representation technique that uses attention mechanisms to learn user and item embeddings, allowing the model to prioritize relevant interactions and improve recommendation quality. Similarly, PGE [63] employs graph embeddings to capture relationships between users and items within a knowledge graph, thereby enhancing the accuracy of recommendations.

Building on the foundations of ID-based and non-ID-based representations, **Multi-dimensional Representation** integrates various aspects of items, users, and interactions. For example, DIN [562] combines these representation techniques to comprehensively capture user and item profiles. Similarly, SRGNN [21] constructs a versatile representation that synthesizes information across items, users, and session dynamics, providing a holistic approach to recommendation modeling. The representation can be formulated as:

$$\mathbf{r}_{\text{multi}} = f(\mathbf{e}_{\text{item}}, \mathbf{e}_{\text{user}}, \mathbf{e}_{\text{session}}) \tag{2.4}$$

where f is a function that integrates item, user, and session embeddings. This approach is particularly beneficial in sequential recommendation, as it allows the model to consider the sequence of user interactions, the context of each interaction, and the intrinsic attributes of items.

Compressed Representation is crucial for managing large-scale datasets by reducing the dimensionality of input data. Techniques such as autoencoders or matrix factorization methods are employed to distill essential information into a more compact form, facilitating faster computations and potentially improving model performance by addressing the curse of dimensionality. For example, the encoding function of an autoencoder can be described as:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}) \tag{2.5}$$

where $\mathbf{x}$ is the input data and $\mathbf{z}$ is the compressed representation. In sequential recommendation, compressed representations can capture underlying patterns in user behavior, aiding in the prediction of future interactions. CpRec [143] exemplifies this approach by using block-wise adaptive decomposition to approximate

input and softmax matrices, exploiting the long-tailed distribution of items in SRS.

Selecting the appropriate representation inductive bias is crucial for the efficacy of sequential recommendation techniques. As research advances, it is vital to explore innovative representations for items, users, and side information to enhance the understanding of their relationships and improve recommendation quality.

2.4.4 Architecture Inductive Biases

Architecture inductive bias refers to the assumptions made by a model's architecture or algorithms about the underlying structure or patterns in the data. These assumptions guide the model's learning process, affecting its ability to generalize from the training data to make accurate predictions on unseen data. In the context of sequential recommendation, the choice of architecture inductive bias can greatly influence the model's capacity to capture temporal dynamics and other essential relationships between items, users, and other side information.

Non-Neural Inductive Biases Despite the rise of neural models, non-neural inductive biases remain crucial for sequential recommendation. They offer benefits such as reduced computational overhead and memory usage. A summary of commonly used techniques is provided below.

Popularity-Based Methods are fundamental recommendation techniques that leverage the popularity of items to generate recommendations. The core principle is that items frequently interacted with by many users are likely to interest other users. The primary inductive bias of these methods is the assumption that users' preferences are significantly influenced by item popularity, suggesting that popular items are generally preferred by most users.

In popularity-based methods, the primary parameter is the popularity metric, typically calculated as the number of times an item has been interacted with or its frequency in user interactions. This metric can be represented as a simple count, a weighted score, or a normalized value. Mathematically, the popularity score of an item i is defined as:

$$\text{Popularity}(i) = \frac{\text{Number of interactions with } i}{\text{Total number of interactions}} \tag{2.6}$$

Common approaches include global item popularity, user-specific item popularity, and session-specific item popularity.

Global item popularity recommends items based on their overall popularity across all users, calculated as:

$$\text{Popularity}_{\text{global}}(i) = \frac{\text{Number of interactions with } i \text{ by all users}}{\text{Total number of interactions by all users}} \tag{2.7}$$

User-specific item popularity recommends items based on their popularity among a subset of users with similar preferences, calculated as:

$$\text{Popularity}_{\text{user-specific}}(i) = \frac{\text{Number of interactions with } i \text{ by similar users}}{\text{Total number of interactions by similar users}} \tag{2.8}$$

Session-specific item popularity recommends items based on their popularity within the current user session, calculated as:

$$\text{Popularity}_{\text{session-specific}}(i) = \frac{\text{Number of interactions with } i \text{ in the current session}}{\text{Total number of interactions in the current session}} \tag{2.9}$$

These methods are often employed as baselines for sequential recommendation models due to their simplicity and effectiveness in specific scenarios. They offer a straightforward benchmark for evaluating the performance of more complex models. Additionally, popularity-based methods can provide recommendations for new users (addressing the cold start problem) or items with limited interaction data, offering a practical solution when interaction data is sparse. An example of a popularity-aware sequential recommendation model is PAUDRec[563], which integrates a popularity-effect module into the main transformer-based model to enhance the performance of the sequential recommendation system.

In conclusion, while popularity-based methods may lack personalization, they are valuable for establishing a foundational performance metric and serving as a fallback mechanism in sequential recommendation systems. They also enhance the main model by integrating popularity effects.

Sequential Pattern Mining (SPM) [564] is a data mining technique aimed at discovering frequent subsequences in a set of sequences. In the context of sequential recommendation, SPM can be utilized to find patterns in user interaction sequences and recommend items based on these patterns. The main inductive bias of SPM is the assumption that users' preferences are influenced by the patterns in their past interactions. Various sequential recommendation studies [565] have employed SPM to extract frequent itemsets or sequential patterns from user interaction data, generating recommendations based on these findings.

Several key techniques in SPM include Association Rule Mining [566], Frequent Pattern Growth (FP-Growth) [567], and PrefixSpan [568].

Association Rule Mining involves identifying interesting associations and relationships between itemsets in transactional data. The key parameters in Association Rule Mining are support, confidence, and lift. Support measures the popularity of an itemset, defined as:

$$\text{Support}(I) = \frac{\text{Number of transactions containing } I}{\text{Total number of transactions}} \tag{2.10}$$

Confidence measures the likelihood of an association, calculated as:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \tag{2.11}$$

Lift evaluates the strength of an association rule by comparing the observed confidence to the expected confidence under the assumption of independence, defined as:

$$\text{Lift}(A \rightarrow B) = \frac{\text{Confidence}(A \rightarrow B)}{\text{Support}(B)} \tag{2.12}$$

The inductive bias of Association Rule Mining is the assumption that frequent itemsets and high-confidence rules are likely to be interesting or useful for predicting future transactions. In sequential recommendation, this technique helps identify patterns and associations in user interaction data, enhancing recommendation accuracy. An example of Association Rule Mining in sequential recommendation

is found in the work of Caser [11], which extracts sequential patterns from user interaction data and generates recommendations based on these patterns.

Frequent Pattern Growth (FP-Growth) is a method used to mine frequent itemsets without candidate generation. It employs a tree structure, called an FP-tree, to represent the dataset in a compact form, allowing for efficient mining of frequent patterns. The FP-tree is constructed by first scanning the dataset to determine the frequency of each item, and then a second scan is conducted to build the tree. Each path in the FP-tree represents a frequent pattern. An equation to describe this process can be:

$$\text{FP-Tree}(T) = \sum_{t \in T} \text{PrefixPath}(t) \tag{2.13}$$

where T is the transaction database and PrefixPath(t) represents the path from the root to the item in the transaction t. The inductive bias of FP-Growth is the assumption that frequent patterns in the dataset can reveal valuable insights for future behavior. In the context of sequential recommendation, this means that the frequent patterns extracted from user interactions can be used to predict future interactions and preferences. FP-Growth is exemplified in the work of [569], where frequent itemsets are extracted from user transaction data to identify common purchasing patterns and recommend items based on these patterns.

PrefixSpan mines the complete set of sequential patterns by examining only the prefix subsequences and projecting their corresponding postfix subsequences into projected databases. The algorithm starts with an initial prefix, which is then recursively extended by exploring frequent items in the projected database. An equation to describe this process can be:

$$\text{ProjectedDatabase}(P) = \{\text{Suffix}(s) \mid s \in S, P \text{ is a prefix of } s\} \tag{2.14}$$

where P is the prefix, S is the sequence database, and Suffix(s) is the suffix of sequence s after removing the prefix P. The inductive bias of PrefixSpan is that patterns in prefix subsequences can be extended to predict future interactions. This bias assumes that understanding the initial parts of sequences can provide insights into their likely continuations. An example of PrefixSpan in sequential

recommendation is presented by [570], where sequential patterns are extracted from user interaction sequences to generate recommendations based on these patterns.

By applying these techniques to sequential recommendation, the discovered patterns and associations can significantly enhance the accuracy and relevance of recommendations provided to users.

Markov Chains (MCs), introduced by Andrey Markov in his seminal work [571], are a foundational choice for developing sequential recommendation models. A Markov Chain is a stochastic process that models a sequence of events where the probability of each event depends solely on the state of the previous event. This characteristic is known as the Markov property, which asserts that the future state of a system is dependent only on the current state, not on the sequence of events that preceded it. The key parameters of a Markov Chain are the states and the transition probabilities between these states.

Typically, a Markov Chain refers to a first-order Markov Chain, where the future state depends only on the current state. Higher-order Markov Chains extend this by assuming that the future state depends on the current state and the previous k states. The complexity of the model and the data required to learn it increase with the value of k, which is usually kept within a practical range (e.g., 2-5). The fundamental inductive bias of Markov Chains is that the future state depends only on the current state, not on the entire sequence of prior states.

Mathematically, a first-order Markov Chain is defined as:

$$P(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_{n+1} = x \mid X_n = x_n) \quad (2.15)$$

In a higher-order Markov Chain, the probability depends on the last k states:

$$P(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_{n+1} = x \mid X_{n-k+1} = x_{n-k+1}, \ldots, X_n = x_n) \quad (2.16)$$

Higher-order Markov Chains and Hidden Markov Models (HMMs) extend the basic Markov Chain model by considering multiple previous states or incorporating hidden states that influence the observed states. HMMs were introduced in the work [572]. An HMM is defined by a set of states, a transition probability matrix between states, and an emission probability matrix that relates the hidden states to the observed events. The probability of a sequence of observations $O = (o_1, o_2, \ldots, o_T)$ given a sequence of states $S = (s_1, s_2, \ldots, s_T)$ is defined as:

$$P(O \mid S) = \prod_{t=1}^{T} P(o_t \mid s_t) P(s_t \mid s_{t-1}) \tag{2.17}$$

where $P(o_t \mid s_t)$ is the emission probability of observation o_t given state s_t, and $P(s_t \mid s_{t-1})$ is the transition probability between states s_{t-1} and s_t. The dependencies in Markov Chains are typically linear, but more complex models can introduce non-linear dependencies.

The inductive bias of Markov Chains, which assumes limited memory, offers both advantages and disadvantages. This assumption simplifies the learning process by reducing the amount of historical information needed to predict future events. In the context of sequential recommendation, this bias means that predictions are based on recent user interactions rather than the entire interaction history, which can be both beneficial for efficiency and limiting in capturing long-term dependencies.

Some weaknesses of the Markov Chain's inductive bias include limited memory, which may fail to capture long-term dependencies and complex patterns; the stationarity assumption, which presumes that transition probabilities remain constant over time; scalability issues, as the size of the transition matrix grows quadratically with the number of states; and difficulty in modeling higher-order dependencies, which may require more sophisticated models.

Despite these limitations, Markov Chains remain valuable due to their simplicity and efficiency in learning. Representative works include FPMC [1], which combines Markov Chains with Matrix Factorization. Fossil [2] addresses FPMC's limitation by integrating similarity-based methods with longer Markov Chains. HCM [3] uses a semi-Markov model to capture active and inactive user states. SPMC [4] combines social and sequential information to enhance recommendation performance.

DEEMS [5] incorporates Markov Chains as one of the choices for its sequential units.

These advancements demonstrate the evolving nature of sequential recommendation systems, leveraging the strengths of various methodologies to improve the accuracy and relevance of recommendations.

Factorizations, such as matrix factorization [573], tensor factorization and Possion factorization are widely used in sequential recommendation systems. They aim to decompose the user-item interaction matrix into lower-dimensional matrices to capture latent factors that influence user preferences and item characteristics. The primary inductive bias of factorization techniques is the assumption that a limited set of latent factors can effectively model user-item interactions.

Matrix factorization stands as a pivotal collaborative filtering method in recommendation systems, breaking down a vast user-item interaction matrix into more manageable user and item latent factor matrices. Its main goal is to unearth latent factors that account for the noticeable user-item interactions, be they ratings or clicks. The dimensionality of the latent factor matrices—usually far smaller than the initial matrix—and the regularization components that mitigate overfitting are the central parameters in matrix factorization. Mathematically, matrix factorization aims to decompose the interaction matrix R into two lower-dimensional matrices U (user factors) and V (item factors):

$$R \approx U \cdot V^T \tag{2.18}$$

where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$, with k being the number of latent factors.

The inductive bias in matrix factorization involves two main assumptions: the low-rank assumption, which posits that a user-item interaction matrix can be replicated using the multiplication of two matrices with reduced dimensions, suggesting that a limited set of latent factors can effectively dictate user preferences and the characteristics of items; and the assumption of linear relationships, which presupposes that interactions can be predicted by linearly combining user preferences with item attributes. In the context of sequential recommendation, this inductive bias implies that the system assumes a relatively small number of factors are sufficient to explain user interactions over time, and these interactions can be modeled linearly.

For sequential recommendation, various extensions and adaptations of factorization techniques have been proposed. Tensor Factorization, as seen in ALMM [112], utilizes a tensor factorization approach, leveraging content features derived from music audio to address the cold start issue in next-song recommendation. This method's inductive bias assumes that considering multi-dimensional interactions (such as user, item, and context) can provide a richer representation of user preferences. Contextual Basket Factorization, exemplified by CBFM [84], is a factorization-based model that incorporates various types of associations involving the user, the target item to be recommended, and the items currently in the basket, specifically for next basket recommendation. The inductive bias here is that user preferences can be inferred from the context of current basket contents and their interactions with other items. Recurrent Poisson Factorization, introduced in RPF [176], treats time as a natural constituent of the model and introduces a rich family of time-sensitive factorization models using Poisson factorization to account for temporal dynamics in user behavior. This method's inductive bias assumes that user interactions follow a Poisson process and that temporal dynamics are crucial for accurate recommendations.

Other factorization method are also commonly seen in this domain. Common examples include Singular Value Decomposition (SVD) [574] and Probabilistic Matrix Factorization (PMF) [575]. Bayesian approaches such as Bayesian Personalized Ranking (BPR) [576] and Bayesian PMF (BPMF) [577] add probabilistic elements to these models. Enhancements to basic techniques include SVD++ [578] and Non-negative Matrix Factorization (NMF) [579]. Additionally, Factorization Machines (FM) [580] are notable for their versatility in modeling interactions within the data.

Overall, factorization techniques provide a robust framework for capturing latent structures in user-item interactions. Their application in sequential recommendation allows for the modeling of evolving user preferences and the integration of contextual and temporal information, enhancing the accuracy and relevance of the recommendations.

k-Nearest Neighbors (kNN) is a non-parametric, instance-based machine learning algorithm used for classification and regression tasks, which can also be adapted for recommendation systems. The key idea behind kNN is to find a predetermined

number (k) of training samples closest in distance to a new input point and predict the label or value based on these nearest neighbors.

Key hyper-parameters include k, the number of nearest neighbors considered for making predictions, the distance metric used to calculate the distance between data points, and optionally, the weights which determine how much influence each neighbor has on the final prediction. Different distance metrics used in kNN include Euclidean Distance:

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.19}$$

Manhattan Distance:

$$d_{\text{Manhattan}}(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.20}$$

and Cosine Similarity:

$$\text{Cosine}(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.21}$$

For distance-based weighting, the weight w_i of the i-th neighbor can be inversely proportional to the distance:

$$w_i = \frac{1}{d(x, x_i)} \tag{2.22}$$

The inductive bias in kNN includes the smoothness assumption, which assumes that instances close together in the input space are more likely to have similar output values, and the bias-variance trade-off. The choice of k determines the trade-off: a smaller value of k results in a more flexible model with higher variance and lower bias, while a larger k leads to a smoother model with lower variance and higher bias.

In the kNN algorithm process using Euclidean distance, the value of k, the number of nearest neighbors, is first determined. For a given user u and item i, the

Euclidean distance between u and all other users based on their interactions is calculated as:

$$d_{\text{Euclidean}}(u, v) = \sqrt{\sum_{j=1}^{m} (r_{uj} - r_{vj})^2} \tag{2.23}$$

where r_{uj} and r_{vj} are the ratings or interactions of users u and v for item j. The k nearest neighbors to user u are then identified based on the calculated distances. The rating or interaction for user u and item i is predicted based on the average ratings or interactions of the k nearest neighbors:

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_k(u)} r_{vi}}{k} \tag{2.24}$$

where $\mathcal{N}_k(u)$ is the set of k nearest neighbors to user u, and r_{vi} is the rating or interaction of neighbor v with item i.

In sequential recommendation, kNN can be adapted to consider the temporal dynamics of user interactions. The distance metric can be modified to incorporate the time between interactions, allowing the model to capture more recent user preferences. For example, a time-weighted distance metric could be used:

$$d_{\text{time-weighted}}(x, y) = \sqrt{\sum_{i=1}^{n} \alpha^{t_i} (x_i - y_i)^2} \tag{2.25}$$

where α is a decay factor and t_i represents the time difference for each interaction.

While many sequential recommendation methods have evolved, traditional methods like kNN remain influential. For instance, STAN [134] not only encompasses basic stacked kNNs but also considers factors such as item position within the current session, the recency of past sessions in relation to the present, and the position of a recommended item in an adjacent session. Additionally, TIFU-KNN [154] utilizes kNN to identify averaged neighbors for each target user, combining collaborative and repeated purchase components to make final predictions.

Overall, kNN provides a straightforward and interpretable approach to recommendation, particularly useful in scenarios where capturing local patterns and similarities is crucial. Its ability to adapt to different distance metrics and weighting schemes makes it versatile for various recommendation tasks.

Translation-based Methods, specifically in the context of recommendation systems, are a family of methods inspired by translation models in natural language processing. The core idea is to represent items and users (or item sequences) in a latent space and model the relationships between them as translations. One well-known translation-based method for recommendation is TransRec.

Key hyper-parameters for translation-based methods include the embedding dimension, which determines the dimensionality of the latent space where users, items, or item sequences are represented; the learning rate, which is the step size used for optimization during the training process; and the regularization strength, which is the weight of the regularization term to prevent overfitting.

The inductive bias of translation-based methods involves several key assumptions. Firstly, these methods assume that the relationships between items (or item sequences) and users can be represented as translations in a latent space. This means that each user and item can be mapped to a point in a high-dimensional space, and user-item interactions can be modeled as vector translations. Secondly, translation-based methods assume that similar items or users are close in the latent space. This proximity allows the model to capture the underlying patterns in user preferences and item characteristics by linearly combining their embeddings. In the context of sequential recommendation, the inductive bias implies that the transitions between consecutive items in a user's interaction sequence can be effectively modeled as translations in the latent space. This helps in capturing the sequential dependencies and temporal dynamics of user behavior, leading to more accurate and relevant recommendations.

In TransRec [88], users are embedded as translation vectors through a shared item embedding space. The model represents user-item interactions as:

$$\mathbf{u} + \mathbf{i} \approx \mathbf{j} \tag{2.26}$$

where **u** is the user embedding, **i** is the embedding of the current item, and **j** is the embedding of the next item in the sequence. TransFM [118] combines the distance and translation components of the TransRec model with the ability of Factorization Machines (FMs) [580] to incorporate arbitrary real-valued features for sequential recommendation. The model can be represented as:

$$\mathbf{u} + \mathbf{i} + \sum_{f \in F} w_f x_f \approx \mathbf{j} \tag{2.27}$$

where F is the set of additional features, w_f are the feature weights, and x_f are the feature values. HierTrans [173] extends traditional item-level relations to the category-level, helping capture dynamic sequence patterns that can generalize across users and time. The model incorporates hierarchical relationships by embedding both items and their categories:

$$\mathbf{u} + \mathbf{i} + \mathbf{c} \approx \mathbf{j} \tag{2.28}$$

where **c** is the category embedding.

Translation-based methods provide a powerful framework for modeling sequential recommendation by capturing the transitions between items in a latent space. They leverage the geometric properties of the latent space to model user preferences and item relationships effectively. These methods are particularly suited for scenarios where capturing the sequential nature of user interactions is crucial for providing accurate and relevant recommendations.

In summary, the inductive bias of translation-based methods for sequential recommendation is that user-item interactions and the transitions between items in a sequence can be represented as translations in a latent space. This bias allows these models to effectively capture the sequential dependencies and temporal dynamics in user behavior, enhancing the model's ability to predict future interactions based on past behavior. The flexibility of translation-based methods to incorporate additional features and hierarchical relationships further strengthens their capability to provide accurate and contextually relevant recommendations.

Graph-based Methods in recommendation systems use graph structures to depict the complex interconnections between users and items, as well as item sequences. Representing users and items as nodes and their interactions as edges, these techniques utilize graph algorithms to infer preferences and identify sequential trends.

Key hyper-parameters for graph-based methods include the graph structure, which determines the type of graph used to represent the data (bipartite, directed, undirected, etc.); node and edge features, which are the attributes associated with users, items, and their interactions; neighborhood size, which is the number of adjacent nodes considered for neighborhood-based methods; and layers and hidden units, which pertain to the depth and width of the graph neural network architecture.

The inductive bias of graph-based methods involves the assumption that the relationships between users, items, and their interactions can be represented using graph structures. These methods assume that users and items with similar interaction patterns are likely to have similar preferences or exhibit similar sequential patterns. This bias helps in capturing the underlying patterns in user behavior and item characteristics by leveraging the graph structure.

Graph-based methods require a built graph to be calculated on, and here, graph-based embedding and random walk are used for sequential recommendations.

In PGE [63], the graph-based embedding is learned by incorporating all the item information into a product graph first, and then using a non-neural method that incorporates time-decay functions to obtain the embeddings. The embeddings are represented as:

$$\mathbf{e}_i = f(\mathbf{A}, \mathbf{X}, t) \tag{2.29}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{X}$ is the node feature matrix, and t is the time-decay function.

Another commonly used technique based on graphs is the random walk. In mathematics, a random walk is also known as a stochastic or random process. It describes a path that consists of a succession of random steps on some mathematical space. In sequential recommendation, random walk techniques can be used to explore the

graph structure and capture the sequential dependencies between items. For example, [99] integrates random walk to address the grocery shopping problem, recommending the next basket of items to users. Using item random walks, S-Walk [302] utilizes the random walk with restart to fully capture intra- and inter-correlations of sessions. The random walk process can be represented as:

$$P(v_{t+1} = j \mid v_t = i) = \frac{A_{ij}}{\sum_k A_{ik}} \tag{2.30}$$

where $P(v_{t+1} = j \mid v_t = i)$ is the probability of transitioning from node i to node j, and A is the adjacency matrix.

Graph-based methods leverage the rich relational information in the graph structure to provide accurate and contextually relevant recommendations. By capturing the complex interactions between users and items, these methods can effectively model user preferences and sequential patterns, enhancing the overall performance of recommendation systems. In summary, the inductive bias of graph-based methods assumes that the complex relationships in user-item interactions can be effectively represented using graph structures, which helps in capturing the inherent patterns in user behavior and item characteristics, leading to more accurate recommendations.

Rule-based Systems rely on manually crafted rules to generate recommendations. These systems use domain knowledge and predefined heuristics to create if-then rules that determine the recommendations based on user behavior and item attributes.

Key hyper-parameters for rule-based systems include rule complexity, which refers to the number and complexity of the rules used, and rule evaluation criteria, which are the metrics used to evaluate the effectiveness of the rules.

The inductive bias of rule-based systems involves the assumption that domain knowledge and simple heuristics can effectively capture user preferences and generate relevant recommendations. These systems work well in domains where expert knowledge is available and the recommendation criteria are well understood. This bias implies that the explicit rules crafted based on domain knowledge can adequately represent the relationships between users and items, leading to accurate recommendations.

In the context of sequential recommendation, rule-based systems can be particularly useful for capturing and utilizing temporal patterns in user behavior. For instance, rules can be designed to recommend items based on the sequence and timing of a user's previous interactions. This approach can be effective in scenarios where there are clear, predictable patterns in user behavior over time.

For example, a rule-based system for a video streaming service might have rules such as:

$$\begin{aligned} &\text{IF user watches a movie from genre X,} \\ &\text{THEN recommend another movie from genre X next.} \end{aligned} \tag{2.31}$$

$$\begin{aligned} &\text{IF user watches a new release,} \\ &\text{THEN recommend the latest releases in the same genre.} \end{aligned} \tag{2.32}$$

These rules leverage temporal patterns in user behavior to make sequential recommendations. By considering the order and timing of user interactions, rule-based systems can provide recommendations that are not only relevant but also timely.

Another example might involve recommending products in an e-commerce setting based on recent purchase sequences:

$$\text{IF user purchases item A, THEN recommend item B if it is often bought next.} \tag{2.33}$$

or:

$$\text{IF user views item X, THEN recommend items frequently viewed afterwards.} \tag{2.34}$$

In these examples, the rules capture the sequential nature of user interactions, allowing the system to recommend items that align with the user's recent activity and anticipated needs.

Rule-based systems are particularly useful in early-stage recommendation systems or in domains with well-defined user preferences and behaviors. They provide a solid foundation for more complex models and offer interpretable recommendations based on explicit rules. Additionally, they can be easily adapted to capture sequential patterns by updating or adding new rules that consider the order and timing of user interactions.

In summary, rule-based systems in sequential recommendation leverage the inductive bias that explicit rules based on domain knowledge can model the temporal dynamics of user behavior. By capturing the sequence and timing of interactions, these systems can provide accurate and relevant recommendations, making them an effective and interpretable approach for initial recommendation tasks and in well-defined domains.

Neural inductive biases refer to the assumptions made by neural network architectures about the underlying structure or patterns in the data. These biases guide the learning process of neural networks and influence their ability to generalize from the training data to make accurate predictions on unseen data. In the context of sequential recommendation, the choice of neural inductive bias can greatly impact the model's capacity to capture temporal dynamics, long-term dependencies, and other essential relationships between items, users, and other side information.

Multilayer Perceptrons (MLPs) [52] are a class of feedforward neural networks consisting of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer. MLPs have the advantage of being able to approximate any measurable function, and their architecture is simple and concise.

In the realm of sequential recommendation, MLPs are utilized to capture sequential patterns in user interactions. For example, NNRec [27] features an embedding layer, a hidden layer, and a softmax output layer, effectively capturing sequential patterns for next basket recommendations. Similarly, HRM [26] captures both sequential behavior and users' general tastes by involving transaction and user representations through different aggregation operations, particularly nonlinear operations.

An advanced model, ListNetSIE [64], employs a list-wise deep neural network to capture the constrained user behaviors evident within individual sessions, effectively modeling dependencies and constraints within session-based interactions. AutoMLP [499] leverages MLP architectures to achieve competitive performance against state-of-the-art methods on both open-source benchmark datasets and real-world industrial applications. This model also features an automated short-term interest length search algorithm, enhancing efficiency and performance. FMLP-Rec [315] is an all-MLP model with learnable filters designed for sequential recommendation tasks. The MLP architecture endows this model with lower time complexity, and the learnable filters, optimized by SGD, adaptively attenuate noise information in the frequency domain. These filters are equivalent to circular convolution in the time domain, providing a larger receptive field and better capturing periodic characteristics.

Additionally, MLPs often serve as preceding layers for prediction, ensuring dimensionality consistency in more complex models. Given the ubiquity of such approaches, enumerating all relevant works might be superfluous here.

Key hyper-parameters for MLPs include the number of layers, the number of neurons per layer, activation functions (e.g., ReLU, sigmoid, tanh), the learning rate, and the regularization strength. These parameters influence the model's ability to capture complex patterns and prevent overfitting.

The inductive bias of MLPs assumes that the relationships between inputs and outputs can be captured through multiple layers of nonlinear transformations. They also assume that complex patterns can be learned through hierarchical feature extraction. This inductive bias makes MLPs particularly effective in capturing complex relationships and dependencies in sequential data, making them suitable for various sequential recommendation tasks.

Overall, MLPs are a fundamental component of neural network architectures, demonstrating versatility and effectiveness in capturing user preferences and predicting future interactions in sequential recommendation systems. By stacking multiple layers and neurons, MLPs can model intricate patterns in user interactions, leading to more accurate and personalized recommendations.

Recurrent Neural Networks (RNNs) are undoubtedly the most intuitive type of neural network for sequence modeling, due to their success in areas including

text, speech, and time-series data. The inductive bias behind RNNs is that they assume an order dependency between the previous and current items in a sequence, where the order can be temporal or some pre-set nature, and use a recurrent structure to encode this dependency.

In the context of sequential recommendation, advanced variants such as Long Short-Term Memory (LSTM) [581] and Gated Recurrent Units (GRUs) [582] have been shown to be effective in capturing long-term dependencies between items in a user's interaction history. These models can generate the next item that the user is most likely to interact with. One advantage of RNNs in this context is their ability to handle variable-length sequences of items, which is useful in modeling user behavior.

Key hyper-parameters for RNNs include the number of layers, the size of the hidden state in each layer, the learning rate, the weight of the regularization term to prevent overfitting, and the batch size, which is the number of sequences processed in parallel during training.

The inductive bias of RNNs assumes an order dependency between the previous and current items in a sequence, which is crucial for capturing sequential patterns. They are designed to handle sequences where the order of interactions is important, making them suitable for time-series data and sequential recommendation tasks.

Mathematically, an RNN processes a sequence of inputs $x_1, x_2, \ldots, x_T$ by iteratively updating its hidden state h_t using the following equations:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b_h) \tag{2.35}$$

$$y_t = W_y h_t + b_y \tag{2.36}$$

where W_h, W_x, W_y are weight matrices, b_h, b_y are bias vectors, and σ is an activation function such as $tanh$ or $ReLU$.

Several sequential recommendation systems have been proposed based on RNNs. For example, GRU4Rec [6] was the first solution for session-based recommendation, and GRU4Rec+ [111] improved it by adding data augmentation and accounting for

shifts in the input data distribution. GRU4Rec++ [65] introduced novel ranking loss functions tailored to RNNs in the recommendation setting. DREAM [7] proposed a dynamic representation to capture global sequential features among baskets for next basket recommendation. RRN [8] endowed both users and movies with an LSTM autoregressive model to capture dynamics and handle time prediction. NSR [155] merged survival analysis with an LSTM to gauge when a user might revisit a site and predict their forthcoming behavior. HRNN [9] personalized RNN models with cross-session information transfer in a hierarchical style, and THRNN [10] added a point process to it, enhancing its power. BINN [103] used LSTMs for discriminative behavior learning after obtaining neural item embeddings, and KrNN [130] accommodated multiple sequences jointly, modeling their interactions through a K-plet recurrent neural network.

Various types of recurrent units have been designed for sequential recommendation problems. For example, TimeLSTM [83] equips LSTM with time gates to model time intervals. π-Net [133] integrates a Shared Account Filter Unit (SFU) and a Cross-Domain Transfer Unit (CTU) to address shared account complexities and combine temporal insights for cross-domain transitions. DCNSR [72] designs a Contextual Gated Recurrent Unit (CGRU) network to account for actions like "click", "collect", and "buy". HLN [148] designs a Leap Recurrent Unit (LRU) to skip preference-unrelated items and utilize previously learned preferences. LRURec [477] introduces linear recurrence with matrix diagonalization to efficiently capture user transition patterns and a recursive parallelization framework that significantly accelerates training and inference.

Overall, RNNs are a powerful tool for capturing sequential patterns in user interactions and generating accurate recommendations. Their ability to model long-term dependencies and temporal dynamics makes them well-suited for sequential recommendation tasks. The advanced RNN-based models effectively capture both short-term and long-term user preferences, leading to more accurate and personalized recommendations.

Convolutional Neural Networks (CNNs) are powerful in processing grid-like data through convolution and pooling operations. Sequential interactions can be transformed into grid-like structures, allowing CNNs to extract compact features from them.

Mathematically, a CNN processes a sequence of inputs $x_1, x_2, \ldots, x_T$ by applying a series of convolutional operations:

$$h_t^l = \sigma(W^l * h_t^{l-1} + b^l) \tag{2.37}$$

where h_t^l represents the output of the t-th position in the l-th layer, W^l is the convolutional filter, b^l is the bias term, $*$ denotes the convolution operation, and σ is an activation function such as $ReLU$.

Key hyper-parameters for CNNs include the number of filters, the size of the convolutional filters, the stride and padding, the size and type of pooling operations (e.g., max pooling, average pooling), the number of layers, the learning rate, and the regularization strength. These parameters influence the model's ability to capture complex patterns and prevent overfitting.

The inductive bias of CNNs assumes that local patterns are important, which aligns with the idea that sequential dependencies can be captured through local interactions. Additionally, through convolution and pooling operations, CNNs assume that patterns can appear anywhere in the sequence, which is useful for detecting features irrespective of their position. Moreover, CNNs build hierarchical representations from low-level to high-level features, capturing complex patterns and interactions in sequential data.

Several works have approached sequential recommendation from a CNN perspective. CASER [11] conceptualizes a sequence of recent items as an "image" in both latent and time dimensions, discerning sequential patterns by treating them as local features of this image. CosRec [12] encodes a sequence of items into a three-way tensor, learns local features using 2D convolutional filters, and aggregates high-order interactions in a feed-forward manner. 3DCNN [13] leverages 3-dimensional CNNs and character-level encoding for input data preprocessing, capturing spatio-temporal patterns while reducing the need for extensive feature engineering across diverse data types. PeterRec [152] adopts a transfer learning methodology by first pretraining to model user interaction dependencies using an autoregressive method, which is a stack of dilated convolutional layers, and then performs fine-tuning using a causal CNN network. NextItNet [14] utilizes Temporal Convolutional Networks (TCNs) [583] to efficiently increase the receptive fields and add causal constraints,

adopting a residual architecture [584] to ease the optimization of deeper networks. HierTCN [15] implements a low-level model with TCN, utilizing both long-term interests and short-term interactions within sessions to predict the next interaction. CTRec [131] uses a CNN to capture short-term demands. HMN [71] leverages a densely connected CNN with shortcut paths between upstream and downstream convolutional blocks to build multi-scale features from item representations. CpRec [143] adopts an adaptive block-wise decomposition for the input sequences, processes the embeddings through stacked TCN layers (NextItNet), and passes them through another block-wise embedding layer for final recommendation. GRec [170] employs a two-way TCN for encoding item sequences by considering future contexts, using a seq2seq setting with a gap-filling method, and applying causal CNNs as the decoder for the final results. GHTID [487] proposes a graph convolution module based on the global item-item co-occurrence graph and a graph convolution module based on the local item-item transition graph, learning heterogeneous item transition relationships at the global and local levels.

CNNs are effective in extracting features from sequential data due to their ability to model local dependencies and hierarchical structures. By transforming sequential interactions into grid-like structures, CNNs can leverage their powerful feature extraction capabilities to improve the accuracy and relevance of sequential recommendations.

Attentions are mechanisms that allow neural networks to focus on specific parts of the input data while processing it. From Seq2Seq [585] to different variants including soft or hard attention [586, 587], co-attention [588], visual attention [589], hierarchical attention [590], self-attention of Transformers [30], bi-directional self-attention in BERT [591], and bertology-related architectures, attention mechanisms have shown great success in tasks across domains of NLP and CV. By attending to different parts of data with different weights, the varying influence of different parts on the target can be better represented, boosting overall performance. In sequential recommendation, the sequential nature of the data makes it amenable to applying any type of attention, leading to many variations of this powerful mechanism for the problem.

Key hyper-parameters for attention mechanisms include the number of attention heads used to capture different aspects of the input, the size of the embedding vectors used in the attention mechanism, the dropout rate applied to attention

weights to prevent overfitting, the learning rate, and the weight of the regularization term to prevent overfitting.

The inductive bias of attention mechanisms assumes that the relevance of different parts of the input can be weighted differently, allowing the model to focus on the most relevant parts. By dynamically adjusting attention weights, these mechanisms can capture flexible dependencies within sequences. Hierarchical attention mechanisms can capture different levels of relationships within the input data.

Mathematically, an attention mechanism computes a weighted sum of input values, where the weights are determined by a compatibility function between a query and key. For an input sequence $x = \{x_1, x_2, \ldots, x_n\}$, the attention output z is given by:

$$z_i = \sum_{j=1}^{n} \alpha_{ij} v_j \tag{2.38}$$

where α_{ij} are the attention weights computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{2.39}$$

and e_{ij} is the compatibility score, often computed as:

$$e_{ij} = \text{score}(q_i, k_j) \tag{2.40}$$

where q_i is the query, k_j is the key, and v_j is the value.

Several works have approached sequential recommendation using various attention mechanisms. NARM [61] utilizes a hybrid encoder bolstered by an attention mechanism to capture a user's sequential actions and prevailing intent in the current session. EDRec [116] incorporates attention to explicitly learn the expressive portions of sequences, improving performance. DTCN [592] incorporates both temporal context and temporal attention. ATEM [56] uses attention to weight each observed item in a transaction without assuming order. ANAM [29] adopts attention to model the user's evolving appetite for items, using a hierarchical architecture to incorporate attribute information for next basket recommendation.

SRGNN [21] uses soft attention to aggregate global preferences and interests of the current session. Sets2Sets [28] decodes the set of elements at each subsequent time step from vectors using a set-based attention mechanism. ISLF [95] captures the user's main intention by considering both long and short-term interests and latent factors. SSRM [105] improves attention mechanisms by leveraging users' historical interactions for streaming session-based recommendation. RepeatNet [58] uses modified attention for repeat recommendation decoding and item-level attention for explore recommendation decoding. MARank [59] uses attention to obtain a unified embedding, maintaining individual-level interactions through a linear combination of mapped item features. OAR [161] captures a global occasion signal using an attention layer to model seasonal or recurring occasions. GCE-GNN [25] builds layers based on GCN and GAT for global and session-level item representations, using soft attention for the final session representation. IMfOU [168] utilizes node-level attention to capture users' intentions regarding different neighbors and aggregates information for target nodes. CAR [159] employs a consistency-aware attention mechanism as part of its architecture. MSSR [485] designs the multi-sequence integrated attention layer to adaptively leverage both intra-sequence and inter-sequence interaction to learn a user's multiple representations. MAN [480] with local and global modules, mixing three attention networks and transferring at the group level. The first one was the local/global encoding layer that captures the sequential pattern from domain-specific and cross-domain perspectives. Secondly, we further proposed the item similarity attention that captured the similarity between local/global item embeddings and target item embedding, the sequence-fusion attention that fused sequential patterns across global encoder and local encoder, and the group-prototype attention with several group prototypes to share the sequential user behaviors implicitly without leveraging the user ID. Atten-Mixer [475] designs an efficient and effective SBR model with multi-level reasoning component based on mixture of attention through reflecting on the human reasoning process. DFAR [489] consists of feedback-aware encoding layer, dual-interest disentangling layer and prediction layer. In the feedback-aware encoding layer, each head of multi-heads attention can capture specific feedback relations. Factorization-heads attention masks specific head interaction and injects feedback information to factorize the relation between different types of feedback.

Hierarchical attention [590] effectively captures hierarchical relationships within data. Due to hierarchical features in sequential recommendation tasks, it is widely

applied. SHAN [87] uses a two-layer hierarchical attention network for sequential recommendation. The first attention layer learns long-term preferences based on historical purchases, while the second layer outputs final user representation by coupling long-term and short-term preferences. RNS [92] encodes users or items using aspect-aware representations derived from reviews and captures sequential patterns at both union and individual levels using hierarchical attention.

Mathematically, hierarchical attention mechanisms can be represented as a combination of multiple attention layers. For a two-layer hierarchical attention, the output of the first attention layer is fed into the second attention layer, allowing the model to capture higher-level representations:

$$h_i^{(1)} = \sum_{j=1}^{n} \alpha_{ij}^{(1)} v_j \tag{2.41}$$

$$h_i^{(2)} = \sum_{k=1}^{m} \alpha_{ik}^{(2)} h_k^{(1)} \tag{2.42}$$

where $\alpha_{ij}^{(1)}$ and $\alpha_{ik}^{(2)}$ are the attention weights for the first and second layers, respectively.

Co-attentions take two sequences as input to find the relational importance across them. DCNSR [72] employs a co-attention network to capture the dynamic interplay between a user's long-term and short-term interaction behaviors, resulting in a co-dependent representation of both enduring and immediate interests. SCoRe [163] leverages co-attention to discern relationships across neighbors in both direct interactions and co-interactions sets.

Mathematically, co-attention mechanisms compute attention weights for two sequences simultaneously. Given two sequences x and y, the co-attention weights and outputs can be computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{2.43}$$

$$\beta_{ij} = \frac{\exp(f_{ij})}{\sum_{k=1}^{m} \exp(f_{ik})} \tag{2.44}$$

$$z_i^x = \sum_{j=1}^{m} \alpha_{ij} y_j \tag{2.45}$$

$$z_i^y = \sum_{j=1}^{n} \beta_{ij} x_j \tag{2.46}$$

where α_{ij} and β_{ij} are the co-attention weights, and e_{ij} and f_{ij} are the compatibility scores for the sequences x and y.

Memory networks [593] function as cascaded attentions with storing, reading, and writing capabilities, resembling real memories. They are utilized in sequential recommendation tasks. RUM [156] employs a memory-augmented neural network to explicitly store and update users' historical interactions, enhancing model expressiveness. KSR [129] fuses RNN-based architectures with the Key-Value Memory Network (KV-MN) and incorporates knowledge base information for enhanced semantic representation. HPMN [136] adopts a hierarchical and periodical updating mechanism to capture multi-scale sequential patterns of user interests. MIMN [594] is a memory-based architecture capturing user interests from long sequential behavior data. KAMemNN [162] schedules embeddings of users, items, and categories into a memory net to extract vital information. MAN [98] augments a base neural recommendation with a continuously queried and updated key-value nonparametric memory. CSRM [132] incorporates two parallel memory modules to apply collaborative neighborhood information to session-based recommendations. ICMSR [146] incorporates memory nets in the intent-guided retrieval module.

Mathematically, memory networks use attention mechanisms to read from and write to memory slots. Given an input sequence x and a memory matrix M, the attention weights and memory read operation can be computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} \tag{2.47}$$

$$r_i = \sum_{j=1}^{m} \alpha_{ij} M_j \tag{2.48}$$

where α_{ij} are the attention weights and e_{ij} are the compatibility scores between the input and memory slots.

Self-Attention and Transformers focus on the relations within a sequence itself, where each pair of items receives relative weights. SASRec [16] is a self-attention-based sequential model with positional encoding based on relatively few actions. AttRec [17] utilizes self-attention under a metric learning framework. CSAN [595] projects heterogeneous user behaviors into a common latent semantic space and feeds the output into a self-attention network. ATRank [57] projects heterogeneous user behaviors into multiple latent semantic spaces, with influence among behaviors via self-attention. FDSA [18] applies self-attention blocks on item-level and feature-level sequences to model item and feature transition patterns. LINet [80] integrates self-attention with Gaussian weighting to extract sub-session features. BST [596] harnesses the Transformer model to discern sequential patterns in user behavior sequences at Alibaba. RKSA [19] adopts customized self-attention with relations from items, users, and global co-occurrence information through a kernel function. GCSAN [94] applies self-attention on session representations for long-range dependencies. SDM [597] encodes behavior sequences with a multi-head self-attention module and a long-short term gated fusion module. STRSA [598] learns relationships among threads by applying self-attention. MFGAN [142] uses a Transformer-based generator and factor-specific discriminators. MRIF [144] uses transformers to extract user interests and then aggregates them with various types of aggregators. CoSAN [97] uses kNN for collaborative item representation with a Transformer for final results. TiSASRec [20] captures both the absolute positions of items and the intervals between successive items using self-attention. ASLI [172] employs self-attention to discover similarities among items, combined with TCN layers. DSNTSP [31] uses a multi-head self-attention layer and an item-oriented attention layer. ESRM-KG [171] encodes action sequences into high-dimensional representations through multi-head self-attention. CTA [169] uses dynamic calibration to fine-tune the relative input dependence from self-attention. SelCa [70] is a self-attentive sequential recommendation system with topic modeling-based category embedding. GAG [150] uses a self-attention layer and MLP for session-level representation. HyperRec [138] uses hypergraph convolution to extract item correlations and self-attention for dynamic preference. GLSGRL [137] uses GNN for long and short-term preferences and self-attention for group representation. SGS [151] integrates a dual-channel fusion mechanism with self-attention for preference

generation. CTRec [131] employs self-attention to capture recurring purchase patterns in long-term demands. TASER [147] uses a Transformer-like mechanism for encoding time intervals among items. DEEMS [5] considers self-attention as one of its choices for sequential units. MQSA-TED [479] introduces an L-query self-attention module using flexible window sizes for attention queries and combined long and short-query self-attentions. STOSA [319] introduces a novel Wasserstein Self-Attention module to characterize item-item position-wise relationships in sequences, effectively incorporating uncertainty into model training. LISA [311], an efficient attention mechanism for recommendation, built upon embedding quantization with codebooks. In LISA, code word histograms for each code book are computed over the input sequences. The histograms and the inner product between code words are then used to compute the attention weights, in time linear to the sequence length. Trans2D [305] presents a novel watchlist recommendation (WLR) task, an extended Transformer model with a novel self-attention mechanism that attends to 2D-array data (item-attribute) inputs.

Mathematically, self-attention mechanisms compute attention weights based on the input sequence itself. Given an input sequence $x = \{x_1, x_2, \ldots, x_n\}$, the self-attention output z is computed as:

$$z_i = \sum_{j=1}^{n} \alpha_{ij} v_j \tag{2.49}$$

where α_{ij} are the attention weights computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{2.50}$$

and e_{ij} is the compatibility score, often computed as:

$$e_{ij} = \text{score}(q_i, k_j) \tag{2.51}$$

where q_i is the query, k_j is the key, and v_j is the value.

The Transformer model extends self-attention by stacking multiple layers of self-attention and feed-forward neural networks. The architecture of the Transformer

includes multi-head self-attention mechanisms, which allow the model to focus on different parts of the input sequence simultaneously. Given an input sequence $x = \{x_1, x_2, \ldots, x_n\}$, the multi-head self-attention mechanism computes multiple sets of attention weights and values. The output z is computed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{2.52}$$

where each attention head head_i is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.53}$$

and the attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.54}$$

where Q, K, and V are the query, key, and value matrices, respectively, W_i^Q, W_i^K, and W_i^V are the projection matrices for the i-th head, and W^O is the output projection matrix.

BERT (Bidirectional Encoder Representations from Transformers) [591] is one of the most advanced bases for NLP tasks, utilizing stacked bi-directional transformers to train models from both left-to-right and reverse directions to use more context information. BERT4Rec [69] employs deep bidirectional self-attention to model user behavior sequences. To avoid information leakage and efficiently train the bidirectional model, the Cloze objective is adopted for sequential recommendation, predicting the random masked items in the sequence by jointly conditioning on their left and right context. This method allows learning a bidirectional representation model to make recommendations by allowing each item in user historical behaviors to fuse information from both sides.

Mathematically, BERT uses the Transformer architecture, which includes multi-head self-attention mechanisms. Given an input sequence $x = \{x_1, x_2, \ldots, x_n\}$, the multi-head self-attention mechanism computes multiple sets of attention weights and values. The output z is computed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{2.55}$$

where each attention head head_i is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.56}$$

and the attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.57}$$

where Q, K, and V are the query, key, and value matrices, respectively, W_i^Q, W_i^K, and W_i^V are the projection matrices for the i-th head, and W^O is the output projection matrix.

Overall, attention mechanisms are powerful tools for capturing complex relationships within sequential data. By focusing on different parts of the input data with varying weights, attention mechanisms can effectively model dependencies and interactions, leading to more accurate and personalized recommendations.

Graph Neural Networks (GNNs) [599, 600] epitomize a specialized neural inductive bias adept at accurately modeling and representing data with a graph structure. Notable advanced iterations include the Graph Convolutional Network (GCN) [601], Graph Attention Network (GAT) [602], and Gated Graph Neural Network (GGNN) [603].

Many models based on GNN and its advanced variants have been built for sequential recommendation tasks, leveraging the graph structure to model complex relationships among items and users. SRGNN [21] utilizes pure GNN to capture the sequential dependencies among session items by assigning each item a node embedding. TAGNN [24] upgrades SRGNN by using GGNN and adds a target attention mechanism to calculate scores for all items in the session with respect to the target. DGRec [71] is based on a dynamic-graph-attention neural network, modeling context-dependent social influence with a GAT that dynamically infers influencers based on users' current interests. FGNN [22] approaches the next item recommendation task through the lens of graph classification, using a weighted attention

graph layer and a Readout function to derive embeddings for items and sessions. GAG [150] transforms the session into a weighted graph with global attributes, performing graph convolution on the graph with node features, edge features, and global attributes. HyperRec [138] uses a hypergraph to model item correlations, connecting multiple items with one hyperedge and employing a hypergraph convolution network to extract both direct and high-order connections as short-term correlations between items. MAGNN [23] uses GNN for short-term interest modeling, a memory net for long-term interest modeling, and a bi-linear function for item co-occurrence patterns. GCE-GNN [25] builds layers based on GCN and GAT for global-level and session-level item representations. GLSGRL [137] uses GNN to represent users' long and short-term preferences based on constructed corresponding graphs. MKMSR [139] uses micro-behavior sequences, which refer to an item and an operation committed on the item, and learns the embeddings via GGNN and GRU. DGNN [474] proposes to decouple the modeling of explicit dependencies and implicit correlations among items for session-based recommendation, where a GNN with a single gate (SG-GNN) captures the explicit dependencies as reflected by the ordering of items in sessions, and an adaptive GNN (A-GNN) learns implicit correlations between any two items adaptively with a self-learning strategy. RetaGNN [313] is a novel graph neural network (GNN) based model, Relational Temporal Attentive GNN (RetaGNN), which achieves its goal without relying on content and auxiliary information by learning relational attentive weight matrices in the enclosing subgraphs centered at the target user-item pair. HG-GNN [307] proposes a heterogeneous global graph neural network for personalized session-based recommendation by considering the impact of historical interactions of users and building a heterogeneous global graph consisting of historical user-item interactions, item transitions, and global co-occurrence information. SERec [300] constructs a heterogeneous knowledge graph from the social network and historical user behaviors, applies a heterogeneous GNN to learn KG embeddings of users and items, and feeds these embeddings into an SR model to provide more accurate recommendations.

Key hyper-parameters for GNNs include the number of graph convolutional layers, the size of the hidden state in each layer, the number of attention heads for attention-based GNNs, the learning rate, and the weight of the regularization term to prevent overfitting.

The inductive bias of GNNs includes relational modeling, where GNNs assume that the relationships between nodes (users and items) can be effectively captured using graph structures; local neighborhood aggregation, where GNNs aggregate information from a node's local neighborhood to learn representations, assuming that nearby nodes have significant influence on each other; and hierarchical feature extraction, where by stacking multiple graph convolutional layers, GNNs can capture hierarchical and high-order interactions within the graph.

Mathematically, a GNN processes a graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. For each layer l, the node embeddings $H^{(l)}$ are updated as:

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)}) \tag{2.58}$$

where A is the adjacency matrix, $H^{(l)}$ is the node embedding matrix at layer l, $W^{(l)}$ is the weight matrix, and σ is an activation function such as $ReLU$.

For attention-based GNNs like GAT, the update rule involves attention coefficients:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[Wh_i \| Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T[Wh_i \| Wh_k]))} \tag{2.59}$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right) \tag{2.60}$$

where α_{ij} is the attention coefficient for edge (i, j), a is a learnable weight vector, and $\|$ denotes concatenation.

GNNs are effective in capturing complex relationships and dependencies in sequential data through their ability to model relational structures. By leveraging graph-based inductive biases, these networks can enhance the accuracy and relevance of sequential recommendations.

Knowledge Graph (KG) [604] represents a collection of interlinked descriptions of entities, which can include real-world objects, events, situations, or abstract concepts. These descriptions have a formal structure that allows both people and

computers to process them efficiently and unambiguously. Entity descriptions contribute to one another, forming a network where each entity represents part of the description of related entities.

Knowledge graphs are used as a tool for sequential recommendation as well. Notable examples include NARE [153], which generates item embeddings through a knowledge graph approach (TransE) [605], then uses LSTM and attention mechanisms for the final prediction. Chorus [140] integrates dynamic items through relational representation and temporal kernel functions in a knowledge-aware and time-aware manner. KSR [129] melds RNN-based architectures with the Key-Value Memory Network (KV-MN), enhancing semantic representation through the inclusion of knowledge base data, augmenting model capabilities and interpretability. KERL [141] performs reinforcement learning (RL) with the fusion of a knowledge graph. TKG-SRec [476] enhances KGs by tracking dynamic user behaviors to create Temporal Knowledge Graphs (TKGs), where the introduced temporal knowledge in TKGs focuses on both entity-related and graph structure-related facets.

Key hyper-parameters for knowledge graph-based models include the embedding dimension, the learning rate, the weight of the regularization term to prevent overfitting, and knowledge graph completion techniques such as TransE, TransH, or RotatE.

The inductive bias of knowledge graph-based models includes entity relationships, where the relationships between entities are captured and used to enhance recommendation models; semantic integration, where integrating knowledge graphs allows models to leverage semantic information to improve recommendations; and contextual understanding, where knowledge graphs enable models to understand the context of items and users by linking related entities.

Mathematically, knowledge graph embeddings represent entities and relations in a continuous vector space. For a triplet (h, r, t) where h is the head entity, r is the relation, and t is the tail entity, the embedding model aims to ensure that:

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t} \tag{2.61}$$

where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ are the embeddings of the head entity, relation, and tail entity, respectively. The objective function for training these embeddings typically minimizes the distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$.

Knowledge graphs are effective in enhancing sequential recommendation models by providing semantic information about entities and their relationships. By leveraging the structured knowledge in the graph, models can make more informed and context-aware recommendations, leading to improved accuracy and relevance.

Adversarial Deep Learning attempts to fool machine learning models by supplying deceptive input, and it is a popular learning paradigm for training models to achieve high accuracy and robustness to adversarial inputs. Generative Adversarial Networks (GANs) [606] are successful generative models incorporating adversarial learning, creating new data instances that resemble the training data. For example, GANs can create images that look like photographs of human faces, even though the faces don't belong to any real person. A GAN has two parts: the generator, which learns to generate plausible data, and the discriminator, which learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

GANs have been applied in various domains, including computer vision and information retrieval, and are also popular in sequential recommendation. Selected works under adversarial learning or GANs include LARA [160], which is based on conditional GANs, using multiple generators to generate users from multiple perspectives of items' attributes and a discriminator. SAO [149] adds adversarial perturbations into SASRec to enhance the generalization of sequential-based factorized approaches. AOS4Rec [243] formalizes the sequential recommendation problem as a seq2seq learning problem, introducing adversarial oracular learning by sequence-level oracle in the auto-regression, with SSRGAN. MFGAN [142] employs transformers both as the main building blocks for a generator and as factor-specific discriminators, performing multi-factor adversarial learning under a reinforcement learning framework. RecGURU [304] is a novel cross-domain sequential recommendation framework based on Generalized User Representations (GURs), where an autoencoder generates user representations in each individual domain. Cross-domain generalization is achieved by adversarially training a discriminator and the encoder until domain-dependent embeddings are statistically indistinguishable among different domains.

Besides GANs, other adversarial learning techniques have been used to improve sequential recommendations. Adversarial training (AT) [607] enhances model robustness by training with adversarial examples. This approach is particularly effective in handling noise and perturbations in user interaction data. Variants of adversarial training include Projected Gradient Descent (PGD) [607] and Fast Gradient Sign Method (FGSM) [608], which generate adversarial examples to improve the robustness of sequential recommendation models.

In the context of adversarial deep learning for sequential recommendation, key hyper-parameters include the generator architecture, the discriminator architecture, the learning rate, the weight of the regularization term to prevent overfitting, and the adversarial perturbation level.

The inductive bias of adversarial learning-based models includes generative modeling, where GANs assume that a generative model can learn to produce plausible data that resembles the training data; adversarial robustness, where training with adversarial examples makes models more robust to deceptive inputs; and improved generalization, where adversarial training helps models generalize better by exposing them to challenging and diverse examples.

Mathematically, GANs are trained by optimizing the following minimax objective function:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.62}$$

where G is the generator, D is the discriminator, x is a real data instance, and z is a random noise vector. The generator G tries to generate data that the discriminator D cannot distinguish from real data, while the discriminator D tries to correctly classify real and generated data.

For adversarial training methods, the goal is to minimize the loss function under adversarial perturbations:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \Delta} \mathcal{L}(f_\theta(x + \delta), y) \right] \tag{2.63}$$

where θ represents the model parameters, (x, y) are the data-label pairs, δ is the adversarial perturbation, and $\mathcal{L}$ is the loss function. Δ denotes the set of permissible perturbations.

Adversarial learning, particularly using GANs and adversarial training techniques, enhances sequential recommendation models by generating realistic user interaction sequences and improving robustness to adversarial inputs. This results in more accurate and resilient recommendation systems.

Autoencoders (AEs) are foundational unsupervised neural network models designed to generate a representation that recovers the original input after encoding it through various transformations. There are several variants of AEs, including Sparse Autoencoder (SAE) [609], Denoising Autoencoder (DAE) [610], and Variational Autoencoder (VAE) [611]. VAEs are particularly noteworthy as generative models used extensively in sequential recommendation systems.

A Variational Autoencoder (VAE) extends the traditional autoencoder by regularizing the training process to avoid overfitting and ensure that the latent space is structured to support generative tasks. VAEs consist of an encoder and a decoder and are trained to minimize the reconstruction error between the encoded-decoded data and the initial input. Unlike standard autoencoders, VAEs encode the input as a distribution over the latent space, introducing regularization to the latent variables.

Key hyper-parameters for VAEs include the latent space dimension, learning rate, regularization strength, number of layers in the encoder and decoder networks, and the weight assigned to the reconstruction loss in the training objective. The inductive biases of AEs involve the assumption that high-dimensional input data can be effectively represented in a lower-dimensional latent space while preserving essential information. VAEs specifically introduce a regularized latent space to ensure it has properties conducive to generating new data instances, and they emphasize reconstruction fidelity, aiming to accurately reconstruct input data.

Mathematically, a VAE is trained by optimizing the following objective function, which combines a reconstruction loss with a regularization term:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p(z)) \tag{2.64}$$

where $q_\phi(z|x)$ represents the encoder approximating the posterior distribution, $p_\theta(x|z)$ denotes the decoder generating the data, and D_{KL} is the Kullback-Leibler divergence between the approximate posterior and the prior distribution $p(z)$.

Several models incorporating VAE for sequential recommendation highlight its versatility and effectiveness. For instance, CVAE [100] builds a framework that integrates rating and content through VAE, learning a latent distribution for content in the latent space rather than the observation space. Similarly, SVAE [158] uses a recurrent version of VAE, feeding a consumption sequence subset through an RNN to model the probability distribution of likely future preferences at each time step. ISLF [95] employs a recurrent VAE to capture both the primary intent and interest fluctuations of users, considering long-term and short-term interests while integrating latent factors. VASER [167] extends VAE to Bayesian inference for sequential recommendation, introducing normalizing flows to better estimate the probabilistic posterior. ACVAE [312] enhances the encoder through adversarial learning via the AVB framework, employing contrastive learning and including a special convolutional layer in the encoder after the recurrent layer to capture short-term sequence information.

VAEs offer a robust framework for modeling complex sequential data, generating realistic user interaction sequences, and capturing intricate user behavior patterns. By leveraging the strengths of autoencoders and variational inference, VAEs improve the accuracy and robustness of sequential recommendation systems.

Reinforcement Learning (RL) is an area of machine learning focused on how agents should take actions in an environment to maximize cumulative rewards. RL methods have achieved significant success in various domains such as robotics and games, with notable applications including AlphaGo and its variants.

Several RL-based works have been developed for sequential recommendation. For example, CDB [121] recommends the next song to users in an online reinforcement learning setting, leveraging implicit feedback as the sole signal of user preference. LOKI [174] utilizes the reinforcement learning algorithm to train an attack agent, generating user behavior samples for data poisoning. KERL [141] integrates knowledge graph information into the RL framework for next item recommendation.

Key hyper-parameters in RL include the learning rate, which is the step size used for optimization during the training process, the discount factor, which discounts

future rewards, the exploration rate, which balances the exploration of new actions versus exploiting known rewarding actions, and the reward function, which calculates the reward for the agent's actions.

The inductive biases of RL in sequential recommendation include the assumption of sequential decision-making, where each action affects future states and rewards, the aim to maximize cumulative rewards over time, aligning the recommendation system's goals with long-term user satisfaction, and the inherent balance of exploration and exploitation, helping to discover new and relevant items for users.

Mathematically, RL optimizes the expected cumulative reward R defined as:

$$R = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{2.65}$$

where r_t is the reward at time step t, γ is the discount factor, and T is the total number of time steps.

In RL for sequential recommendation, the system is modeled as an agent interacting with an environment. The agent's actions (recommendations) influence the environment (user interactions), and the agent receives rewards based on the user's feedback. The objective is to learn a policy $\pi(a|s)$ that maximizes the expected cumulative reward, where a is an action and s is the state.

Several algorithms and techniques are used to train RL models for sequential recommendation. Q-learning is an off-policy algorithm that learns the value of state-action pairs to derive the optimal policy. Deep Q-Network (DQN) extends Q-learning by using deep neural networks to approximate the Q-value function. Policy Gradient Methods directly optimize the policy by gradient ascent on expected reward. Actor-Critic Methods combine value-based and policy-based methods, with an actor updating the policy and a critic evaluating it.

By leveraging reinforcement learning, sequential recommendation systems can adaptively learn user preferences and dynamically adjust recommendations to maximize long-term user engagement and satisfaction. RL-based approaches provide a robust framework for modeling complex user interactions and optimizing recommendations in a sequential decision-making context.

Self-Supervised Learning (SSL) [612] is a learning paradigm where models are trained on the data itself without requiring explicit labels. Instead, models are trained to predict some part of the input data from another part, effectively learning useful representations in an unsupervised manner. SSL has been successfully applied in various domains, including computer vision, natural language processing, and sequential recommendation. SSL techniques often utilize contrastive learning (CL)[613], a popular approach that aims to learn representations by contrasting positive samples (similar) with negative samples (dissimilar). By maximizing the similarity between positive pairs and minimizing the similarity between negative pairs, CL can effectively learn discriminative representations.

Several SSL-based works in sequential recommendation illustrate this paradigm. S4Rec [510] introduces an Online Self-Supervised Self-Distillation (OSSD) framework, combining cluster-aware self-distillation, clustering on the fly, and head-tail adversarial learning. S3Rec [76] adopts a pre-training and fine-tuning strategy with four self-supervised tasks. CL4SRec [232] employs three data-level augmentation approaches to construct positive views, utilizing transformers as sequence encoders with distinct contrastive learning tasks. DSSRec [106] introduces an intent variable to extract mutual information between individual user's past and future interaction sequences. ICSRec [478] leverages clustered latent intent factors and contrastive self-supervised learning to optimize sequential recommendation. UnifiedSSR [508] adopts a dual-branch architecture that encodes product history and query history in parallel, effectively sharing information across scenarios and views while modeling dynamic user intent through self-supervised learning signals. MSSR [485] introduces a user representation alignment module to optimize multiple representations of the same user by leveraging self-supervised signals. ICLRec [318] models latent intent factors from user interactions and fuses them into a sequential recommendation model via a new contrastive SSL objective. CLSR [316] disentangles long and short-term interests for recommendation with a contrastive learning framework. Additionally, DuoRec [306] utilizes feature-level augmentation based on dropout to better maintain semantic consistency between positive views, while CoSeRec [614] employs a contrastive loss to align user sequences with similar interaction patterns, pushing apart sequences with different patterns. DA4Rec [504] discusses the importance of data augmentation in contrastive learning for sequential recommendation. EBM [503] develops a noise-decoupling contrastive

learning approach and a guided training strategy that combines the Hard Noise Eliminator and Soft Noise Filter techniques.

Key components of SSL include pretext tasks designed to generate pseudo-labels from the input data itself, such as predicting masked items or reconstructing corrupted inputs, and encoders like neural network architectures (e.g., transformers, CNNs) used to learn representations from the input data. Contrastive learning plays a crucial role in SSL, where the model learns to distinguish between similar and dissimilar pairs of data points. Various augmentation techniques are applied to the input data to create different views, aiding the model in learning robust representations.

SSL's inductive biases include learning from the data structure itself, assuming that useful patterns and relationships can be learned directly from the structure of the data without requiring explicit labels. SSL focuses on learning high-quality representations that can be transferred to downstream tasks, improving overall model performance. By leveraging the inherent structure of the data, SSL can make effective use of large, unlabeled datasets, reducing the need for extensive labeled data.

Mathematically, SSL optimizes an objective function L defined as:

$$L = \mathbb{E}\left[\sum_{i=1}^{N} \ell(f(x_i), y_i)\right] \tag{2.66}$$

where x_i represents the input data, y_i are the pseudo-labels generated by the pretext tasks, f is the model, and ℓ is the loss function.

In the context of SSL for sequential recommendation, the system learns from user interaction sequences by generating pseudo-labels through pretext tasks. These tasks might involve predicting the next item, reconstructing corrupted sequences, or distinguishing between positive and negative samples. The goal is to learn meaningful representations that can improve recommendation accuracy.

Contrastive learning, a critical component of SSL, optimizes a contrastive loss function L_{CL} defined as:

$$L_{CL} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{2.67}$$

where z_i and z_j are the representations of a positive pair, sim denotes the similarity measure (e.g., cosine similarity), τ is a temperature parameter, and N is the batch size.

In sequential recommendation, techniques such as masked item prediction, next item prediction, and hybrid approaches combining multiple pretext tasks are commonly used to leverage different aspects of the data, improving overall model robustness and performance. By leveraging SSL and contrastive learning, sequential recommendation systems can effectively learn from large volumes of unlabeled data, capturing intricate user interaction patterns and enhancing recommendation performance.

Large Language Models (LLMs) have significantly transformed various artificial intelligence domains, including sequential recommendation systems. Prominent LLMs like GPT-4 [615] and LLaMA [616] have paved the way, while other models such as Claude 3 [617], Gemini 1.5 [618], PaLM 2[619], Reka Core [620], Qwen 2 [621] and many others, though not specifically designed for sequential recommendation, have demonstrated the potential of LLMs in processing sequential data effectively. These models, pre-trained on extensive corpora, come with inherent inductive biases that make them particularly effective for processing sequential data [266, 356, 388, 453, 455, 507, 509, 523]. This subsection examines these biases and their implications for sequential recommendation tasks.

Contextual Understanding from Sequential Data. LLMs are inherently designed to process sequential data, primarily textual content. This design predisposes them to excel at understanding and predicting sequences, making them particularly suitable for tasks involving sequential recommendations. Their capacity to comprehend long-range dependencies in sequences enables them to capture user behavior patterns over time, which is a crucial aspect of effective recommendation systems. To address the limitations of LLMs in extracting useful information from long user behavior sequences, several innovative frameworks have been proposed. For instance, [509] designed the Rella framework, which includes Semantic User Behavior Retrieval (SUBR) and Retrieval-Enhanced Instruction Tuning (ReiT) to improve LLMs' comprehension of lifelong sequential behaviors. These techniques help LLMs to overcome the challenge of incomprehension in long user behavior sequences, even when the context length does not exceed the model's limitations [356, 453]. Additionally, [507] proposed the LLM-TRSR framework,

which segments user behavior sequences and leverages the zero-shot summarization capabilities of LLMs. This framework employs an LLM-based summarizer to encapsulate user preferences through hierarchical and recurrent summarization paradigms. Subsequently, LLM-TRSR utilizes an LLM-based recommender for sequential recommendation tasks, fine-tuning its parameters using Supervised Fine-Tuning (SFT) [622]. For parameter-efficient fine-tuning (PEFT)[623], Low-Rank Adaptation (LoRA) is employed, as introduced by [624].

Transfer Learning and Generalization. Pre-trained LLMs possess a broad understanding of language and user interactions due to their extensive training on diverse datasets. This pre-training acts as an inductive bias, allowing these models to generalize effectively across various domains. In the context of sequential recommendation, LLMs can adapt to specific user preferences and item characteristics with minimal fine-tuning, leveraging their substantial knowledge base. Studies such as those on Personalized Prompt-based Recommendation (PPR) and Rella demonstrate how leveraging pre-trained knowledge enables efficient adaptation to new recommendation tasks with enhanced personalization [505, 509, 523]. These models often utilize foundational models like GPT-4 and LLaMA, which have been trained on vast amounts of textual data, providing a strong foundation for understanding user intent and preferences across different contexts.[505] proposed SLIM, which is a step-by-step knowledge distillation module to transfer the stepby-step reasoning capabilities in recommendation from a larger teacher model to a smaller student model (with approximately 4% of the parameters of the teacher model). This smaller model evolves into a proficient reasoner, which can be directly deployed as a "slim" knowledge generator for sequential recommendation. Consequently, this knowledge can be flexibly integrated with any sequential recommendation backbone and utilized in both ID-based and ID-agnostic scenarios.

Attention Mechanisms and Temporal Dynamics. The attention mechanisms in LLMs, particularly those in Transformer architectures, are essential for modeling the relevance of different parts of a sequence. For sequential recommendations, this capability translates into the ability to weigh the importance of recent versus historical interactions, aligning recommendations with current user interests while considering past preferences. The use of attention mechanisms allows models like LLaMA and its adaptations to focus on relevant user-item interactions dynamically, thereby improving recommendation accuracy [388]. By effectively managing

temporal dynamics, these models can provide more accurate and timely recommendations, enhancing the user experience.

Handling Sparse and Long-Tail Data. LLMs' proficiency in handling sparse data and understanding long-tail distributions is a critical bias for sequential recommendation. This capability arises from their exposure to diverse and extensive datasets during training, enabling them to make accurate predictions even with limited user-item interaction data. Approaches like Multiple Key-value Strategy and LoRec demonstrate the effectiveness of LLMs in dealing with sparse datasets and mitigating the challenges posed by long-tail user behavior, with models like LLaMA and GPT-4 being foundational [356, 453]. This proficiency allows LLMs to recommend niche items or cater to users with uncommon preferences, thus broadening the scope and utility of recommendation systems.

Robustness and Adversarial Resistance. One of the emerging areas in the application of LLMs to recommendation systems is enhancing robustness against adversarial attacks. Methods such as LoRec employ LLMs to improve the robustness of recommendations against poisoning attacks, ensuring the integrity and reliability of the recommendation process. LoRec utilizes LLaMA as its foundation model, highlighting the potential of LLMs to not only provide accurate recommendations but also maintain their performance in adversarial settings [453]. This robustness is crucial for maintaining user trust and the overall effectiveness of recommendation systems in real-world applications.

Challenges and Limitations. Despite these advantages, LLMs also face challenges in the context of sequential recommendations. One significant issue is the black-box nature of these models, which complicates the interpretation of their decision-making processes. Additionally, their extensive training requirements pose challenges regarding computational resources and environmental impact [523]. These challenges necessitate ongoing research to enhance the interpretability, efficiency, and sustainability of LLMs in sequential recommendation systems.

Future Directions. Future research in the adoptions and fusions of LLMs to sequential recommendation systems should focus on addressing the challenges of interpretability, scalability, and sustainability. Exploring methods to enhance the transparency of LLMs, optimize their resource consumption, and improve their environmental impact will be crucial for their widespread adoption. Additionally,

investigating novel architectures and training strategies that leverage the strengths of LLMs while mitigating their limitations will be essential for advancing the field of sequential recommendation. Research into hybrid models that combine LLMs with other machine learning techniques could provide new pathways for improving recommendation accuracy and efficiency. Furthermore, developing techniques to better integrate non-textual data and handle privacy concerns will be vital for the practical implementation of LLM-based recommendation systems.

Others: In addition to the above-mentioned inductive biases, several other inductive biases play a crucial role in shaping the learning process and generalization ability of sequential recommendation models. These biases include:

Seq2Seq self-supervised learning, as seen in [106], formalizes sequential recommendation as a seq2seq self-supervised problem with a mixed loss of seq2item and seq2seq. ComiRec [107] regards multiple user interests as interest capsules and uses dynamic routing from Capsule Networks [625] to extract them. HGN [104] captures feature and instance-related parameters through hierarchical gating mechanisms. Beacon [96] models each basket by encoding the relative importance of items and correlations among item pairs, utilized to infer sequential associations along the basket sequence. BSEQ [89] models the generation of both target and support basket sequences using "Siamese networks" and "fraternal networks" for parameter sharing. GSU [135] models sequential graphs in a spectral domain and applies spectral convolution operations to obtain better results. Additionally, [120] explores the limits of session-based recommendation tasks.

Key hyper-parameters for these architectures include the learning rate, embedding dimension, regularization strength, batch size, and specific parameters for loss functions. For instance, the size of the latent space where inputs are encoded, the depth of the encoder and decoder networks, and the weight given to the reconstruction loss in the training objective are crucial for optimizing model performance.

The inductive biases of these architectures provide the foundational assumptions that guide their design and learning processes. For example, disentangled representation assumes that user interests can be separated into distinct factors or capsules, as in ComiRec. Hyperbolic space modeling, utilized in some models, assumes that embeddings in hyperbolic space can better capture hierarchical relationships and long-term dependencies. Hierarchical gating mechanisms, seen in HGN, assume

that such structures can effectively capture feature and instance-specific parameters. Sequential associations, as in Beacon, assume that encoding item importance and correlations in baskets can improve sequential association inference. Spectral convolution, used in GSU, assumes that modeling sequential graphs in a spectral domain can capture complex dependencies.

Mathematically, these models leverage different structures and transformations to optimize their respective objective functions. Seq2Seq models optimize a sequence-to-sequence mapping using a loss function that captures both seq2item and seq2seq predictions. Capsule Networks (ComiRec) use dynamic routing [625] to cluster and transform input features into higher-level capsules representing user interests. Hyperbolic space models map input sequences into a hyperbolic latent space [626], optimizing embeddings that preserve the hierarchical nature of data.

These diverse architectures offer unique inductive biases and methodological advantages, enabling them to address specific challenges and opportunities in sequential recommendation tasks. By leveraging techniques such as self-supervised learning, capsule networks, hyperbolic embeddings, hierarchical gating, and spectral convolution, these models enhance the capability and performance of recommendation systems.

Overall, the inductive biases and hyper-parameters of these architectures play a crucial role in shaping the learning process and the generalization ability of the models. By understanding and leveraging these biases and parameters effectively, researchers and practitioners can design more effective and robust sequential recommendation systems that cater to the diverse needs and preferences of users.

2.4.5 Objective Inductive Biases

Objective inductive bias in sequential recommendation refers to the assumptions and predispositions made when designing the objective function, which influences the learning process and the model's generalization ability. These biases shape how models' training objective is formulated, what aspects of the data are emphasized, and how the model learns to make recommendations. By carefully selecting and designing the objective function, researchers can guide the model to focus on specific aspects of the data. This section explores various objective inductive biases

commonly used in sequential recommendation models and their implications for model performance and generalization.

Pointwise Objectives treat each instance in the sequence independently, ignoring any dependencies between items. This approach often results in models that fail to capture the inherent sequential nature of the data. For example, in a sequence of user interactions with items, the model would treat each interaction as an isolated event. Examples of pointwise objectives include mean squared error (MSE) and binary cross-entropy loss. Mathematically, the pointwise objective for MSE can be formulated as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.68}$$

where y_i is the actual rating and $\hat{y}_i$ is the predicted rating. Another important objective is the Binary Cross-Entropy loss, which can be expressed as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{2.69}$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability. Both MSE and BCE are commonly used in pointwise recommendation models, but they may not fully capture the sequential patterns and dependencies in the data.

Pairwise Objectives consider the relative ranking of items in the sequence, capturing the relationships between pairs of items. This inductive bias helps models learn the preferences and transitions between items in the sequence. Examples of pairwise objectives include Bayesian Personalized Ranking (BPR)[576] loss and hinge loss. The BPR loss is defined as:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i,j)\in D} \ln \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}) \tag{2.70}$$

where σ is the sigmoid function, $\hat{y}_{u,i}$ is the predicted score for user u and item i, and D is the set of user-item pairs.

Listwise Objectives aim to optimize the ranking of the entire sequence, taking into account the order of items and their relationships within the sequence. This inductive bias leads to models that are more effective at capturing complex sequential patterns. Examples of listwise objectives include ListMLE[627], ListNet[628],

and LambdaMART[629]. The ListMLE loss is defined as:

$$\mathcal{L}_{\text{ListMLE}} = -\sum_{i=1}^{N} \ln P(\pi_i|x) \tag{2.71}$$

where $P(\pi_i|x)$ is the probability of the permutation π_i given the input x.

Temporal Objectives specifically focus on modeling the temporal dynamics in the sequence, emphasizing the importance of time and order in the recommendation process. Temporal objectives can be designed to account for the time intervals between interactions or the specific order of events. Examples of temporal objectives include time-aware ranking loss and temporal cross-entropy loss. A temporal cross-entropy loss can be expressed as:

$$\mathcal{L}_{\text{TemporalCE}} = -\sum_{t=1}^{T} y_t \log(\hat{y}_t) \tag{2.72}$$

where y_t is the true label at time t and $\hat{y}_t$ is the predicted probability.

Diversity and Serendipity Objectives that incorporate diversity and serendipity biases encourage models to provide more varied and unexpected recommendations, improving the user experience and avoiding filter bubbles. For sequential recommendation, diversity can be introduced by ensuring that the recommended items are different from those previously interacted with. An example objective for diversity can be:

$$\mathcal{L}_{\text{Diversity}} = -\sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{sim}(\hat{y}_i, \hat{y}_j) \tag{2.73}$$

where $\text{sim}(\hat{y}_i, \hat{y}_j)$ measures the similarity between items i and j.

Multitask Learning objectives encourage models to learn multiple tasks simultaneously, leveraging the shared information between tasks to improve the performance of each task. In sequential recommendation, this could involve predicting the next item in the sequence while also predicting user preferences or other auxiliary tasks. The overall loss can be a weighted sum of individual task losses:

$$\mathcal{L}_{\text{MTL}} = \sum_{k=1}^{K} \lambda_k \mathcal{L}_k \tag{2.74}$$

where λ_k is the weight for the k-th task and $\mathcal{L}_k$ is the loss for the k-th task.

Metric Learning objectives encourage models to learn a metric space that can be used to measure the similarity between items, which is crucial for sequential recommendation as it helps in identifying similar items that a user might interact with next. Examples include triplet loss or contrastive loss. The triplet loss can be formulated as:

$$\mathcal{L}_{\text{Triplet}} = \sum_{(a,p,n)} \max(0, d(a,p) - d(a,n) + \alpha) \tag{2.75}$$

where a, p, and n are the anchor, positive, and negative samples, respectively, d is the distance function, and α is the margin.

By carefully selecting and designing the suitable objective functions, researchers can guide the model to focus on specific aspects of the data, such as ranking, temporal dynamics, diversity, or multitask learning. These objective inductive biases play a crucial role in shaping the learning process and the generalization ability of sequential recommendation models, ultimately influencing the quality of recommendations provided to users.

2.4.6 Optimization Inductive Biases

Optimization inductive bias in sequential recommendation focuses on the assumptions and predispositions made during the optimization process, which can have a significant impact on the learning and generalization of the models. Some of the commonly encountered optimization inductive biases include:

Optimization Algorithms, like gradient descent, stochastic gradient descent (SGD)[630], Adam[631], RMSprop, or AdaGrad, can introduce different inductive biases in the optimization process. Each algorithm has its own set of hyperparameters and update rules, which can affect the model's convergence, stability, and generalization ability. For instance, the update rule for SGD is:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{2.76}$$

where θ represents the model parameters, η is the learning rate, and $\nabla_\theta \mathcal{L}(\theta_t)$ is the gradient of the loss function with respect to the model parameters at time step t.

Adam, on the other hand, uses adaptive learning rates:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta \mathcal{L}(\theta_t) \tag{2.77}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta \mathcal{L}(\theta_t))^2 \tag{2.78}$$

$$\hat{\theta}_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{2.79}$$

where m_t and v_t are estimates of the first and second moments of the gradients, β_1 and β_2 are hyperparameters, and ϵ is a small constant.

Learning Rate during optimization can directly influence the speed of convergence and the stability of the model. A smaller learning rate results in slower convergence but might lead to more stable solutions, while a larger learning rate can speed up the convergence but may result in unstable solutions. The effect of the learning rate can be seen in the parameter update equation:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{2.80}$$

where η is the learning rate. Adaptive learning rate schedules, such as learning rate decay, can also be employed:

$$\eta_t = \eta_0 \cdot \frac{1}{1 + \lambda t} \tag{2.81}$$

where η_0 is the initial learning rate, λ is the decay rate, and t is the current epoch.

Regularization techniques, such as L1 or L2 regularization, encourages the model to find simpler solutions, often preventing overfitting. Regularization adds a penalty term to the objective function, constraining the model complexity. For L2 regularization, the objective function becomes:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda \|\theta\|_2^2 \tag{2.82}$$

where λ is the regularization strength and $\|\theta\|_2$ is the L2 norm of the model parameters. Dropout is another regularization technique that randomly drops units during training:

$$\text{Dropout}(h) = h \odot \mathbf{r} \tag{2.83}$$

where h is the hidden layer and $\mathbf{r}$ is a random binary mask.

Batch Size during training can introduce a bias towards smoother optimization landscapes. Smaller batch sizes can result in a more stochastic optimization process, which can help escape local minima, while larger batch sizes lead to a more deterministic optimization process, often converging to flatter regions in the loss landscape. The gradient update for a batch size B is:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta \mathcal{L}(\theta_t, x_i) \tag{2.84}$$

where x_i are the samples in the batch. The effect of batch size can also be balanced using mini-batch training strategies.

Early Stopping is a form of regularization that terminates the training process when the performance on a validation set starts to degrade. This can prevent overfitting and introduce a bias towards simpler models with better generalization ability. The early stopping criterion can be expressed as:

$$\text{Stop training if } \mathcal{L}_{\text{val}}(t) > \mathcal{L}_{\text{val}}(t-k) \tag{2.85}$$

where $\mathcal{L}_{\text{val}}(t)$ is the validation loss at epoch t and k is the number of consecutive epochs with no improvement.

Gradient Clipping is used to prevent the exploding gradient problem in which gradients become too large during training. By clipping gradients to a maximum norm, the model maintains stability during optimization:

$$\nabla_\theta \mathcal{L} \leftarrow \frac{\nabla_\theta \mathcal{L}}{\max(1, \frac{\|\nabla_\theta \mathcal{L}\|_2}{c})} \tag{2.86}$$

where c is the clipping threshold.

Learning Rate Schedules adjust the learning rate during training to improve convergence. Common schedules include step decay, exponential decay, and cosine annealing. For example, exponential decay is defined as:

$$\eta_t = \eta_0 \cdot e^{-\lambda t} \tag{2.87}$$

where η_0 is the initial learning rate, λ is the decay rate, and t is the current epoch.

Warm Restarts periodically reset the learning rate to a higher value, which can help the model escape local minima and improve convergence:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\frac{t \mod T}{T}\pi\right)\right) \tag{2.88}$$

where $\eta_{\min}$ and $\eta_{\max}$ are the minimum and maximum learning rates, and T is the period of the restart.

Hyperparameter Tuning involves selecting the best set of hyperparameters for the optimization algorithm. This can be achieved using grid search, random search, or Bayesian optimization:

$$\mathcal{L}_{\text{opt}} = \min_{\theta} \mathbb{E}_{x \sim p(x)}[\mathcal{L}(x, \theta)] \tag{2.89}$$

where $p(x)$ is the data distribution and θ are the model parameters.

Understanding the optimization inductive biases is essential for developing sequential recommendation models that can effectively learn and generalize from the available data. By carefully selecting optimization algorithms, learning rates, regularization techniques, batch sizes, early stopping criteria, gradient clipping, learning rate schedules, warm restarts, and hyperparameter tuning strategies, researchers can guide the optimization process towards more stable and efficient solutions, ultimately improving the performance of the recommendation system.

2.4.7 Evaluation Inductive Biases

Evaluation inductive bias in sequential recommendation refers to the assumptions and predispositions made when selecting evaluation metrics and setting up experimental protocols. These biases can significantly influence the interpretation of a model's performance and its ability to generalize to real-world scenarios. Common evaluation inductive biases in sequential recommendation include accuracy-based, ranking-based, diversity and novelty, and temporal metrics.

Accuracy-based metrics focus on the correctness of the recommendations, evaluating how well the model predicts the items users will interact with. Examples include Recall, Precision, F1-score, Mean Reciprocal Rank (MRR) and Hit Ratio

(HR). These metrics prioritize accuracy but may not fully represent the diverse and serendipitous nature of real-world recommendations.

Recall measures the proportion of actual relevant items that have been correctly identified by the model. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP (True Positives) is the count of items correctly recommended, and FN (False Negatives) is the count of relevant items not recommended. In sequential recommendation, recall ensures that the model captures all potentially relevant items that align with the evolving preferences of the user. For instance, if a user frequently watches sci-fi movies, recall would measure how well the system continues to recommend new sci-fi movies as the user's preferences develop. However, focusing solely on recall can lead to an overwhelming number of recommendations, which might reduce the practical usability and satisfaction of the user experience.

Precision evaluates the relevance of the items recommended, representing the fraction of relevant items among all the recommended items. It is given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where FP (False Positives) represents the count of irrelevant items that were recommended. In sequential recommendation systems, high precision indicates that the recommendations are pertinent to the user's current context, such as recommending the latest episode of a series the user is currently watching. However, high precision without considering recall can result in missing out on less obvious but potentially interesting items.

F1-Score is the harmonic mean of precision and recall, providing a single score that balances both metrics. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In the context of sequential recommendation, the F1-Score offers a balanced measure that accounts for both the accuracy and completeness of the recommendations. For example, in an e-commerce setting, the F1-Score would help in evaluating how

well the system recommends both popular items and niche products that the user might be interested in based on their browsing history.

Mean Reciprocal Rank (MRR) measures the average of the reciprocal ranks of the first relevant item for each user or query. It is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

where $|Q|$ is the number of queries and rank_i is the rank position of the first relevant item for the i-th query. MRR is particularly relevant in sequential recommendation because it reflects the efficiency with which a user can find the first relevant item, such as the first song they are likely to enjoy in a playlist generated based on their listening history.

Hit Ratio (HR) measures the proportion of users for whom the model correctly predicts at least one relevant item in the recommendation list. It is defined as:

$$\text{Hit Ratio} = \frac{\text{Number of Users with at least one relevant item}}{\text{Total Number of Users}}$$

Hit Ratio is essential in sequential recommendation as it captures the model's ability to provide relevant recommendations to users, ensuring that users are not left without any relevant suggestions. For example, in a movie recommendation system, Hit Ratio would evaluate how often the system successfully recommends at least one movie that the user enjoys.

Ranking-based metrics evaluate the effectiveness of a recommendation system by assessing the order in which items are recommended. Normalized Discounted Cumulative Gain (NDCG) measures the gain of an item based on its position in a ranked list, discounting gains at lower ranks. NDCG is defined as:

$$\text{NDCG} = \frac{DCG}{IDCG}$$

where DCG (Discounted Cumulative Gain) is given by:

$$DCG = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

and $IDCG$ (Ideal Discounted Cumulative Gain) is the maximum possible DCG. This metric reflects the reduced user interest in lower-ranked items but depends heavily on the correct assignment of relevance scores. In sequential recommendation, NDCG is crucial as it aligns with the importance of presenting the most relevant items first, adapting to the user's changing preferences over time. For example, in a news recommendation system, NDCG would measure how effectively the system ranks the most relevant news articles at the top of the list.

Mean Average Precision (MAP) averages the precision at each rank where a relevant item is found, integrating both precision and recall into a single metric. MAP is defined as:

$$\text{MAP} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{AP}_i$$

where AP_i is the Average Precision for the i-th query. While MAP incentivizes higher ranking of relevant items, it is sensitive to the number of relevant items and can be opaque to non-experts. In sequential recommendation, MAP helps in evaluating the overall ranking quality, ensuring that relevant items are consistently prioritized throughout the recommendation list. For instance, in a video streaming platform, MAP would assess how well the system ranks a user's favorite genres higher in the recommendation list.

Area Under the ROC Curve (AUC) quantifies the likelihood that a randomly chosen relevant item is ranked higher than a randomly chosen irrelevant item. AUC is defined as:

$$\text{AUC} = \frac{1}{n_+ n_-} \sum_{i=1}^{n_+} \sum_{j=1}^{n_-} \mathbf{1}(r_i > r_j)$$

where n_+ and n_- are the numbers of positive and negative items respectively, r_i and r_j are the ranks of the positive and negative items, and $\mathbf{1}$ is an indicator function. In the context of sequential recommendation, AUC can help to understand the model's ability to differentiate between relevant and irrelevant items over various thresholds. For example, in a book recommendation system, AUC would evaluate how well the system distinguishes between books the user would read and those they would not, across different genres and interests.

Diversity and Novelty Metrics aim to evaluate the range and uniqueness of

recommendations, ensuring users receive varied and unexpected suggestions. Intra-List Diversity (ILD) measures the dissimilarity between items within the same recommendation list, enhancing user experience by providing variety but potentially reducing precision. ILD is typically measured as:

$$\text{ILD} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (1 - \text{sim}(i, j))$$

where N is the number of items in the list and $\text{sim}(i, j)$ is the similarity between items i and j. For sequential recommendations, ILD ensures that the recommendations are not monotonous and cover a broad spectrum of user interests. For instance, a music recommendation system would use ILD to ensure a playlist includes a mix of genres and artists, catering to the user's broad musical tastes.

Coverage assesses the extent to which the system suggests items from the entire available catalog, promoting the discovery of niche items but possibly recommending less relevant items. Coverage is defined as:

$$\text{Coverage} = \frac{|\text{Recommended Items}|}{|\text{Total Items}|}$$

In sequential recommendation, coverage ensures that users are exposed to a wider range of items, including long-tail items that they might not encounter otherwise. For example, in an online marketplace, coverage would measure how well the system recommends not only popular products but also unique items from smaller sellers.

Serendipity measures the system's ability to recommend items that a user will like but is unlikely to discover independently. While this can greatly increase user satisfaction, it is highly subjective and difficult to measure. Serendipity can be quantified as:

$$\text{Serendipity} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(r_i \notin \text{User Profile}) \cdot \mathbf{1}(\text{User Likes } r_i)$$

where N is the number of recommendations, and r_i is a recommended item. In sequential recommendation, serendipity plays a crucial role in surprising and delighting users with recommendations that go beyond their typical preferences. For

example, a movie recommendation system might introduce users to critically acclaimed independent films they wouldn't normally watch, thereby enhancing their viewing experience.

By carefully selecting evaluation metrics and protocols that align with the specific goals and requirements of the recommendation system, researchers can effectively assess the model's performance and its ability to generalize to real-world scenarios. Understanding the evaluation inductive biases is crucial for interpreting the results of experiments and making informed decisions about the design and optimization of sequential recommendation systems.

Selection of K for Top-K Metrics

The selection of the parameter K in Top-K metrics, such as Precision@K, Recall@K, and NDCG@K, is a critical decision in the evaluation of recommendation systems, as it significantly influences the interpretation of the model's performance. The choice of K should align with the application's specific needs and the user's decision-making process. A lower K value focuses the evaluation on the top of the recommendation list, which is most relevant when the cost of examining additional items is high, or when the user is likely to consider only a few recommendations. In contrast, a higher K allows for a broader assessment, capturing the model's ability to recommend a longer list of items that may be of interest to the user.

For instance, in scenarios where users are expected to explore a large number of recommendations, such as browsing a media library, a higher K might be more appropriate. Conversely, in situations where the goal is to identify the single best item, such as in certain e-commerce contexts, $K = 1$ might suffice.

The performance evaluation of a recommendation system can change drastically with different K values, affecting both the precision and recall of the recommendations. Precision typically decreases with an increase in K, while recall may increase. Therefore, the selection of K should be a thoughtful trade-off based on what aspect of the recommendation is most valuable for the end goal.

2.4.8 Deployment Inductive Biases

Deployment inductive biases in sequential recommendation systems refer to the assumptions and design decisions made when integrating recommendation models into real-world applications. These biases significantly influence the system's performance, scalability, and user experience. While many published works [632–635] may not directly address sequential recommendation, they provide valuable insights into the general principles of recommendation deployment. Some common deployment inductive biases in sequential recommendation include:

Scalability is crucial for a recommendation system's ability to manage increasing volumes of data and user interactions without performance degradation. Systems designed with scalability in mind can efficiently process extensive datasets, accommodate a growing user base, and deliver real-time recommendations seamlessly, even under heavy loads. This is particularly important for platforms like YouTube[634], where the recommendation system must handle vast amounts of content and user interactions daily.

Real-Time Recommendations involve dynamically providing personalized suggestions as users interact with the system, necessitating low-latency responses. Models optimized for real-time operations must quickly process user activities, update recommendations instantly, and adapt to evolving user preferences on-the-fly. The importance of real-time processing is evident in applications such as Facebook's ad recommendation system[632], which needs to respond to user actions promptly to maintain engagement.

Cold-Start Problem occurs when the system lacks sufficient data on new users or items to generate accurate recommendations. Addressing this issue involves designing models that can utilize auxiliary information, such as user demographics or item attributes, to make relevant suggestions despite limited interaction data. Netflix's hybrid approach[633], which combines collaborative filtering with content-based methods, exemplifies effective strategies for mitigating the cold-start problem.

Model Interpretability is crucial for understanding and trusting the recommendations generated. Interpretable models facilitate user trust, regulatory compliance, and the identification and correction of model errors or biases. This transparency is essential for both end-users and system administrators, as highlighted

by the need for clarity in LinkedIn's talent recommendation system[635].

Robustness to Adversarial Attacks involves building models that can resist attempts to manipulate or bias the recommendations. These attacks could lead to skewed suggestions, privacy breaches, or exploitation of system vulnerabilities. Effective models must detect and mitigate such threats to maintain the integrity of the recommendations.

Fairness and Bias Mitigation in recommendation systems involves designing models that do not discriminate against any users based on sensitive attributes. Fairness aims to provide equitable recommendations and avoid systemic biases that could disadvantage specific user groups. This is particularly relevant in professional networking platforms like LinkedIn [635], where fair representation of all users is crucial.

Privacy and Data Protection are critical, especially when handling sensitive user information. Models should be built to safeguard user data, comply with data protection regulations, and minimize risks associated with data breaches or unauthorized access, thereby maintaining user trust and legal compliance.

Model Maintenance and Updates are crucial for keeping recommendation systems accurate and relevant. This involves periodic retraining with new data, updating model parameters, and adapting to changes in user preferences or item popularity. Models designed for ease of maintenance ensure sustained performance and relevance over time.

By integrating these deployment inductive biases into the design and implementation of sequential recommendation systems, researchers and practitioners can develop robust, scalable, and user-centric solutions that effectively address the challenges of real-world deployment. These biases guide the development of systems that are not only accurate and efficient but also trustworthy, fair, and privacy-preserving, ensuring a positive user experience and long-term success in real-world applications.

2.5 Conclusion

This literature review provided a comprehensive exploration of sequential recommendation systems, with particular emphasis on the critical role of inductive biases across different stages of system design. Through systematic analysis of top-tier research, including statistical gathering, trend analysis, and survey reviews, the review examined how various types of inductive biases - from data preprocessing and model architecture to evaluation and deployment - fundamentally shape the behavior and performance of recommendation systems. The review highlighted the transformative potential of neural inductive biases in improving system efficacy while addressing key challenges in capturing dynamic user preferences and complex temporal patterns. By elucidating core principles and providing a detailed taxonomy of inductive biases in the recommendation pipeline, this work serves as a foundational guide for future research and innovation in developing more sophisticated sequential recommendation systems. The comprehensive analysis offers valuable insights into how different inductive biases can be effectively leveraged to enhance both model performance and real-world applicability. Furthermore, this review establishes the essential theoretical foundation for this thesis to develop novel solutions addressing critical gaps in sequential recommendation systems, particularly focusing on the central research questions around variable-length sequences, complex item relationships, and temporal dynamics in user-item interactions.

TABLE 2.6: Summary of Sequential Recommendation Tasks, Proposed Models, Their General Inputs and Outputs Forms, and Application Scenarios

Task	**Inputs**	**Outputs**	**Model**	**App Scenarios**
Next-item Recommendation	User's last L sets of items with timestamps	Next Item	QR-Rec	Online retail platforms, Streaming services, Mobile app recommendations, News article suggestions, Personalized marketing
Next-basket Recommendation	Items' transitional relations across consecutive baskets	Next basket of items	TransGAT	E-commerce platforms, Grocery shopping apps, Fashion outfit recommendations, Home improvement stores, Health and wellness products
Next-items Recommendation	Sequences of past item interactions	Next sequence of items	MSQT	Content recommendation, Playlist generation, Book reading sequences, E-learning course recommendations, Fitness training plans
Temporal Sets Recommendation	Sequence of item sets with timestamps	Next sequence of sets of items	TSNT	Subscription boxes, Monthly service platforms, Seasonal product recommendations, Periodic event planning, Multi-stage project management

Chapter 3

Multi-Scale Quasi-RNN (QR-Rec)

3.1 Introduction

The ability to model sequential user-item interactions has become a cornerstone of modern recommendation systems. Early approaches, such as Markov Chains and Recurrent Neural Networks (RNNs), introduced strong inductive biases that effectively captured order information in item sequences. More recent efforts explored Convolutional Neural Networks (CNNs), leveraging local feature extraction to encode short-range dependencies. Despite these advancements, existing models often treat local patterns in isolation or focus on single-scale features within user histories, potentially overlooking interactions occurring at multiple scales. In this chapter, a new neural architecture called *Multi-Scale Quasi-**RNN** for Next Item **Rec**ommendation* (**QR-Rec**) [636] is proposed. QR-Rec integrates multi-scale convolutional filters with the compositional gating functions of quasi-recurrent cells, capturing a richer spectrum of features from sequential data. By applying convolutional filters of diverse widths, QR-Rec effectively extracts union-level signals across different subsequence lengths. These signals are then processed through quasi-recurrent gating, enabling the model to capture utilize more comprehensive dependencies in a single pipeline.

[1]The work in this chapter has been accepted as "Multi-Scale Quasi-RNN for Next Item Recommendation" in *Proceedings of the 8th International Conference on Crowd Science and Engineering (ICCSE)*, 2025

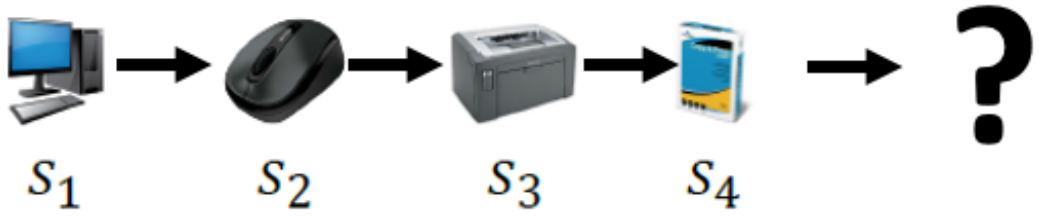


FIGURE 3.1: Next-item recommendation illustration. The temporal ordering of user interactions provides critical clues for predicting future items.

3.2 Problem Definitions and Formulations

The **next item recommendation problem** (Figure 3.1) in sequential recommendation systems is formalized as follows. Consider a sequence of items $S_t^u = \{x_{t-L+1}^u, \ldots, x_{t-1}^u, x_t^u\}$, where x_{t-i+1}^u denotes the item interacted with by user u at time $t - i + 1$. The goal is to predict the subsequent interaction x_{t+1}^u.

The sequence S_t^u reflects a user's interaction history, capturing the temporal dynamics of their behavior. The prediction task is formulated as:

$$\hat{x}_{t+1}^u = \arg \max_{x \in \mathcal{I} \setminus S_t^u} P(x|S_t^u) \tag{3.1}$$

In this formulation, $\hat{x}_{t+1}^u$ represents the predicted next item for user u, $\mathcal{I}$ is the entire item set, and $P(x|S_t^u)$ is the probability of item x being the next interaction, given the sequence S_t^u. This problem is generally approached as a classification task, with the target item x_{t+1}^u as the positive class and all other items in $\mathcal{I} \setminus \{x_{t+1}^u\}$ as negative classes.

Example in Media Streaming: Consider a media streaming platform where user u has watched the following sequence of videos: $S_t^u = \{\text{comedy}, \text{drama}, \text{thriller}, \text{action}\}$. The task is to predict the next genre of video $\hat{x}_{t+1}^u$ the user is likely to watch. Based on the user's viewing history, the model might predict $\hat{x}_{t+1}^u = \text{science fiction}$. By recommending the next genre, the platform can enhance the user's viewing experience by providing content that aligns with their interests.

Example in Online Learning: In an online learning platform, a user u has completed the following sequence of courses:

$$S_t^u = \{\text{Intro to Python}, \text{Data Analysis}, \text{Machine Learning}, \text{Deep Learning}\}$$

The task is to predict the next course $\hat{x}_{t+1}^u$ the user is likely to enroll in. Based on this sequence, the model might predict $\hat{x}_{t+1}^u$ = Reinforcement Learning. This recommendation helps in guiding the user's learning path effectively.

Example in E-commerce: On an e-commerce platform, a user u has the following purchase history: $S_t^u = \{\text{phone}, \text{phone case}, \text{screen protector}, \text{earbuds}\}$. The task is to predict the next item $\hat{x}_{t+1}^u$ the user is likely to buy. Given this history, the model might predict $\hat{x}_{t+1}^u$ = charger. By accurately predicting the next purchase, the platform can improve user satisfaction and optimize inventory management.

3.3 Existing Shortcomings

The GRU4Rec[6] and Caser[11] models, which synergize elements from CNN and RNN architectures, exhibit certain limitations. GRU4Rec, which utilizes a GRU-based RNN approach to adeptly capture user session preferences within item sequences, processes items sequentially, building each state upon its predecessor. However, it may neglect the collective attributes inherent in the entire item sequence. Conversely, Caser employs convolutional filters in two orientations to extract local features from item sequences, and while proficient in capturing localized features and adjacent item relationships, it may not fully capture the ongoing sequential dynamics among these features. To better utilize the strengths of both models, the proposed QR-Rec architecture integrates multi-scale convolutional features within a Quasi-RNN framework. This approach addresses the gap and research question highlighted in Section 1.2 regarding the Fixed-Length Single-Scale User History Limitation and aims to enhance single item recommendation precision.

3.4 Contributions

The principal contributions of this work are delineated as follows, highlighting the novel methodologies and advancements introduced in the QR-Rec model, which address the first research question outlined in Section 1.2: *How to model item order information beyond a single fixed-length user interaction sequence for next-item recommendation?*

- **Innovative Multi-Scale Extension of Quasi-RNN:** This work introduces a novel architecture that enhances Quasi-RNN with dynamic average pooling in a multi-scale framework. By utilizing convolutional filters of varied sizes, the model processes user-item interaction sequence embeddings, extracting diverse union-level features for input into a recurrent structure. This approach significantly improves the learning of sequential representations and enhances the precision of item recommendations.
- **Hierarchical Summation Aggregation:** The Hierarchical Summation Aggregation strategy effectively combines outputs at each architectural level. This method ensures equitable integration of each unit cell and component output, preserving more original information without scaling. Empirical evidence demonstrates that this strategy enhances the model's performance, providing more accurate and reliable results.
- **Empirical Validation and Analysis:** The model's efficacy is validated through comprehensive experiments on 15 real-world datasets from Amazon Reviews. QR-Rec outperforms state-of-the-art methods in metrics such as Mean Average Precision (MAP), Recall@10, and Normalized Discounted Cumulative Gain (NDCG@10). Further in-depth analysis of key hyperparameters and various components underscores the robustness and effectiveness of the proposed model.

Collectively, these contributions significantly advance the field of sequential recommendation systems, demonstrating the effectiveness and robustness of the proposed methodologies in improving recommendation precision.

3.5 Related Work

The advancement of deep learning techniques has significantly influenced various domains, particularly in the context of sequential data processing and recommendation systems. Hybrid CNN-RNN architectures have emerged as a powerful approach, combining the local feature extraction capabilities of Convolutional Neural Networks (CNNs) with the sequential data handling strengths of Recurrent Neural Networks (RNNs). These hybrid models effectively address the limitations of using CNNs or RNNs alone by integrating complex gating mechanisms, which enhance their ability to capture intricate patterns in data. The following subsections delve into specific applications and innovations within the realms of next item recommendation, hybrid CNN-RNN models, and advanced RNN gating mechanisms, highlighting key contributions and developments in each area.

3.5.1 Next Item Recommendation

Next item recommendation, a critical task in sequential recommendation systems, focuses on predicting the subsequent item a user will interact with based on their historical interaction sequence. Traditional methods in this domain utilized collaborative filtering and matrix factorization techniques. However, recent advancements have seen the incorporation of deep learning models, particularly RNNs, to capture sequential dependencies and user preferences more effectively. Variants like LSTMs and GRUs are commonly employed to handle long-term dependencies and mitigate issues like vanishing gradients. Additionally, attention mechanisms have been introduced to enhance model performance by dynamically focusing on relevant parts of the user interaction history. Research such as [6] has shown that session-based RNN models can significantly improve the accuracy of next item predictions by leveraging session-level data, while models like SASRec [16] integrate self-attention mechanisms for even better performance. Building on these foundations, several advanced models have been proposed to address the next item recommendation problem. BERT4Rec [69] utilizes bidirectional encoder representations from transformers to capture sequential patterns by treating the recommendation task as a language modeling problem. Similarly, NextItNet [14] applies dilated convolutions to capture long-term dependencies efficiently, demonstrating superior performance in recommendation accuracy. GNN-based approaches

such as SR-GNN [21] leverage graph structures to model complex item transitions, enhancing recommendation quality. Another notable model is Caser [11], which uses convolutional neural networks to embed a sequence of recent items into an image-like representation, capturing both point-wise and union-level sequential patterns. Hierarchical models such as HGN [104] use hierarchical gating networks to capture both long-term and short-term preferences, while reinforcement learning approaches like DEERS [72] dynamically adjust recommendations based on user feedback and exploration-exploitation trade-offs. These innovations collectively contribute to the robustness and accuracy of next item recommendation systems, making them indispensable in real-world applications. As research in this area continues to evolve, the integration of diverse deep learning architectures and external knowledge sources promises to further enhance the effectiveness of next item recommendation models.

3.5.2 Hybrid CNN-RNN Models

The amalgamation of CNNs and RNNs in neural architectures brings together the best of both worlds: the local feature extraction capabilities of CNNs and the sequential data handling of RNNs. For instance, [637] introduced Long-term Recurrent Convolutional Networks (LRCNs) for visual tasks, effectively combining CNNs for image feature extraction and LSTMs for sequence modeling. In speech recognition, the CLDNN model by [638] integrates CNNs, LSTMs, and deep neural networks, applying each for specific parts of the recognition task. The field of object recognition saw the introduction of Recurrent Convolutional Neural Networks (RCNNs) by [639], which embed recurrent connections within convolutional layers for improved recognition capabilities. In text modeling, [640] developed a hybrid conv-RNN model, integrating a bidirectional RNN with convolutional layers for enhanced language processing. Further advancements in hybrid architectures have been proposed across various domains, such as the modified CNN-RNN hybrid architecture by [641], which focuses on effective training through efficient initialization using synthetic data, image normalization for slant correction, and domain-specific data transformation for learning important invariances. Additionally, [642] proposed an integrated hybrid CNN-RNN model for visual description and the generation of captions, showcasing its versatility. In the realm of music

classification, [643] demonstrated the efficacy of hybrid models in audio processing tasks. Moreover, the Quasi-RNN by [644] utilizes CNN-generated gates for effective pooling in natural language processing, serving as an inspiration for many contemporary studies. These advancements highlight the potential of hybrid CNN-RNN architectures in capturing both spatial and temporal dependencies, leading to significant improvements in various applications such as image recognition, speech processing, text modeling, and music classification.

3.5.3 RNN Gating Mechanisms

The evolution of RNN architectures has seen the development of various gated structures, such as the LSTM [581] and GRU [582], designed to mitigate challenges like vanishing gradients and to improve memory and learning capabilities. The Quasi-RNN [644] employs a unique approach using single-scale k-gram CNNs for adaptive gating, enhancing its flexibility and effectiveness. The Simple Recurrent Unit (SRU) [645] combines feed-forward networks with sigmoid activation gates to streamline training processes. Another notable development is the Recurrently Controlled Recurrent Network (RCRN) [646], which uses recurrent networks to learn gating functions, addressing some of the inherent drawbacks of traditional RNNs while aiming to boost accuracy and training efficiency in sequential tasks. Additionally, the Position-Gated Recurrent Neural Networks (PG-RNN) model by [647] integrates global and local information dynamically for aspect-based sentiment classification by considering aspect word position information. [648] introduced a novel complex gated recurrent cell that combines complex-valued and norm-preserving state transitions with a gating mechanism. Furthermore, [649] extended restricted orthogonal evolution RNNs with a gating mechanism similar to GRU RNNs, incorporating a reset gate and an update gate. These advancements in RNN gating mechanisms highlight the continuous innovation in this field, significantly enhancing the performance and efficiency of RNNs in handling sequential tasks across various applications.

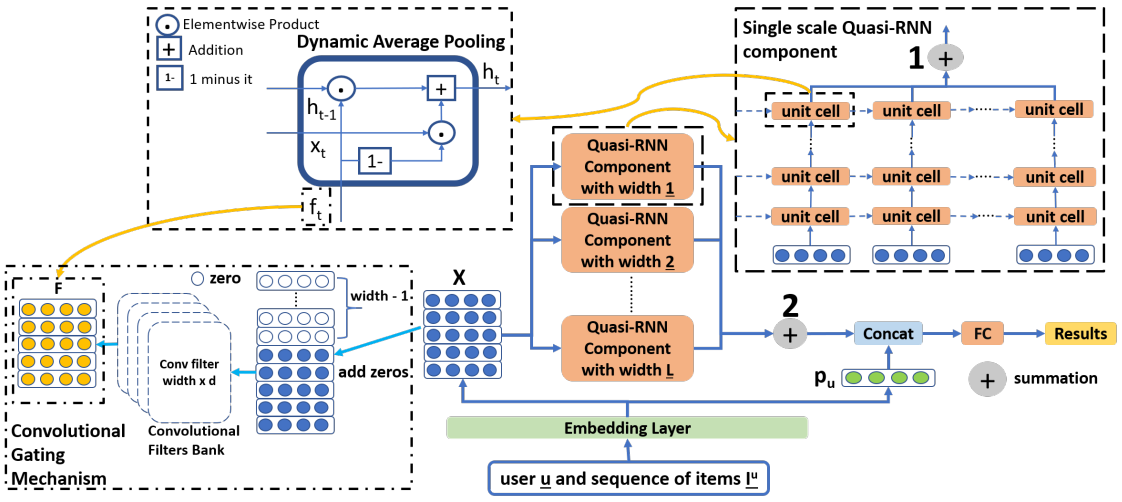


FIGURE 3.2: The Multi-Scale Quasi-RNN (QR-Rec) Architecture

3.6 Methodology

The QR-Rec architecture in Figure 3.2, as detailed in this section, offers an innovative enhancement to next item recommendation by integrating the strengths of both convolutional and recurrent neural network methodologies. This integration is achieved through multi-scale convolution-based features used as recurrent gating functions within the QR-Rec model. The architecture employs multi-scale convolution to pre-learn these gating functions and then applies them in an auto-regressive manner, characteristic of recurrent models. This hybrid approach enables QR-Rec to capture immediate dependencies through its convolutional layers and extended sequential patterns akin to recurrent neural networks, effectively addressing the complexities and temporal dynamics of sequential user-item interactions in recommendation systems.

3.6.1 Embedding Layer

The embedding layer transforms each item in a sequence of length L into a vector in a d-dimensional latent space. Consequently, each item x^u_{t-i+1} is represented in $\mathbb{R}^d$ for $i \in [1, 2, ..., L]$ at time step $t - i + 1$. The output X^u_t in $\mathbb{R}^{d \times L}$ is a sequence embedding matrix, concatenating the embeddings of all items in user u's sequence before time t. User profiles are similarly encoded into this latent space, resulting in a user profile representation $p_u \in \mathbb{R}^d$.

3.6.2 Multi-Scale Quasi-RNN

The core of QR-Rec, the multi-scale Quasi-RNN, applies convolution on input sequences to extract union-level features across different scales. This architecture comprises several subcomponents:

Convolution Gating Mechanism: This mechanism employs convolution to generate gates for the pooling layer, allowing parallel computations. For a filter width of 1, the formula is:

$$F_t^u = \sigma(W_{f,t}^u * X_t^u) \tag{3.2}$$

where $W_{f,t}^u$ is the convolution filter bank, and $*$ indicates masked convolution. Filters of larger width extract higher-level features, leading to a generalized form for gates of width N:

$$f_t^{u,N} = \sigma(W_{f,t}^{u,1} x_{t-N+1}^u + \ldots + W_{f,t}^{u,N} x_t^u) \tag{3.3}$$

Dynamic Average Pooling: This approach uses a forget gate to mix states across time steps independently on each channel:

$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{x}_t \tag{3.4}$$

The pooling component, devoid of trainable parameters, allows for efficient parallel computation.

Hierarchical Summation Aggregation: Aggregation within QR-Rec occurs both at each Quasi-RNN scale and the module's entirety through summation:

$$\mathbf{h}_{w,t}^u = \sum_{t_w=1}^{L} h_{t_w}^u \tag{3.5}$$

$$\mathbf{o}_t^u = \sum_{w=1}^{L} h_{w,t}^u \tag{3.6}$$

3.6.3 Prediction Layer and Recommendation

The user profile p_u is concatenated with output $\mathbf{o}_t^u$ and fed through a fully-connected layer to compute item scores:

$$y_t^u = W_{s,t}^u[\mathbf{o}_t^u; p_u] + b_{s,t}^u \tag{3.7}$$

For recommendations, the top N items with the highest scores from y_t^u are selected. Following the precedent in [578], along with studies like [650] and [16, 651], 100 items from the negative set are randomly sampled alongside the test item for ranking.

3.7 Experiment Setup

The effectiveness of the QR-Rec model was assessed through extensive experiments and detailed qualitative analyses. These were structured to address several pivotal research questions, each aimed at uncovering different facets of QR-Rec's performance and contributions:

RQ1 - How does QR-Rec compare with established baseline models in terms of performance? This inquiry seeks to evaluate whether QR-Rec outperforms these models and, if so, the extent of its improvement in effectiveness.

RQ2 - What factors substantially influence QR-Rec's performance? The goal here is to pinpoint and scrutinize the key elements impacting the results. This analysis will offer deeper insights into the dynamics driving the model's performance.

RQ3 - What is the specific contribution of each component within QR-Rec? This question focuses on dissecting the individual roles and impacts of different model components, aiming to elucidate the mechanisms underlying QR-Rec's effectiveness.

To thoroughly address these questions, the experiment design incorporates various datasets, baselines, and metrics. The datasets span diverse domains, ensuring that the findings are robust and generalizable. Baselines include both traditional and state-of-the-art models in sequential recommendation, providing a comprehensive benchmark for comparison. Performance metrics such as Mean Average Precision

(MAP), Recall, and Normalized Discounted Cumulative Gain (NDCG) are used to evaluate the models, offering a multifaceted view of their effectiveness. Additional analyses focus on the impact of hyperparameters, the contribution of individual model components, and ablation studies to further understand QR-Rec's performance nuances.

3.7.1 Datasets

TABLE 3.1: Statistics of the datasets

Dataset	#users	#items	#interactions	sparsity
Electronics	1530	29381	101436	99.77%
Tools / Home	2908	9565	45786	99.84%
Cell Phones	2440	8152	36246	99.82%
Beauty	4204	11164	75103	99.84%
Health / Care	7398	17257	138083	99.89%
Video Games	5048	10344	96344	99.82%
Toys / Games	3723	11259	65877	99.84%
Sports / Outdoors	6915	17337	109581	99.91%
Pet Supplies	3096	7854	46418	99.81%
Digital Music	1551	3510	35991	99.34%
Grocery / Food	3786	8424	74656	99.77%
Instant Video	540	1527	8133	99.01%
Home / Kitchen	2247	19181	72781	99.83%
Android Apps	2541	10103	81749	99.68%
Baby	3488	6617	53144	99.77%

This study employs **15** diverse datasets Table 3.1 from **Amazon Reviews**, known for their high sparsity, to demonstrate the efficacy of QR-Rec. These datasets are highly suitable for next item recommendation tasks due to their comprehensive representation of user-item interactions across various product categories. The interactions naturally form order sequences, which are exactly suited for the task. Amazon's platform provides a rich source of sequential data where customers review and rate products, capturing the order and timing of interactions. This temporal aspect is crucial for next item recommendation, as it allows the model to learn patterns from the sequences of user activities. The high sparsity of these datasets also presents a realistic challenge, making the evaluation of recommendation models

more robust and reflective of real-world scenarios. In the analysis, the review text indicates user-item interactions, referenced in several studies [651–653].

To ensure the relevance and quality of the datasets, the following criteria were applied:

- **Interactions with user ratings of 3 or above are considered:**This sets a baseline for user preference, ensuring that the interactions considered reflect positive user experiences, which are more indicative of actual user interests.
- **Chronological order is determined using time-related attributes:** This is crucial for next item recommendation tasks as it helps in understanding the sequence in which items were interacted with, allowing the model to predict future interactions based on past behavior.
- **Users with less than 10 interactions are excluded**: [1, 11, 651, 652, 654, 655], This focuses on lengthier user sequences, providing richer data for training and mitigating the cold start problem [11], which is a common issue in recommendation systems. This criterion ensures that the model can learn more meaningful patterns from substantial user histories.

The diversity and size of these 15 datasets provide sufficient support for the research, negating the need for data from other platforms.

3.7.2 Baselines

QR-Rec is benchmarked against **7**established baselines available at the time of this research, each offering a unique approach from non-machine learning to deep learning models, from non-sequential models to advanced sequential models. These baselines are chosen to provide a comprehensive evaluation of QR-Rec's performance across various dimensions, ensuring a robust comparison.

- **PopRec:** This model recommends items purely based on their overall popularity, serving as a simple yet effective benchmark. It reflects general trends but lacks personalization, making it a useful comparison for more sophisticated models that incorporate user-specific preferences.

- **BprMF:** This work [576] utilizes Matrix Factorization within a Bayesian framework to rank items. It emphasizes personalization by focusing on pairwise item comparisons, which is relevant for establishing as a non-sequential baseline.

- **FMC:** This work applies Matrix Factorization on a Markov transition matrix, capturing sequential dependencies between consecutive items. This is particularly important for next item recommendation tasks, as it directly models the sequence of user interactions.

- **FPMC:** The work [1] combines FMC's sequential modeling with general user preferences, enhancing the prediction of the next item. Originally used for next basket recommendation, each basket here contains only a single item for this setup to fit next item recommendation use cases. This combination of sequential and general preference modeling provides a comprehensive benchmark for evaluating QR-Rec.

- **Fossil:** This work[2] is a hybrid model that employs a Similarity Model to capture high-order Markov chains, modeling complex sequential patterns. This allows for a deeper comparison of how QR-Rec handles complex sequential relationships for next item recommendation.

- **GRU4Rec:** This model [6] uses Gated Recurrent Units (GRUs) for session-based recommendation. It is adept at capturing short-term dependencies and session-based dynamics, making it an ideal benchmark for evaluating QR-Rec's ability to model ordered item sequences for next item recommendation.

- **Caser:** The model [11] employs convolutional neural networks (CNNs) to model high-order Markov Chains. It excels in capturing localized sequential patterns in user-item interactions, providing a strong benchmark for assessing QR-Rec's performance in capturing localized sequence information, given the sequence nature of inputs to be extracted for the next item recommendation.

Each of these models brings distinct strengths to sequential recommendation, offering a comprehensive baseline for evaluating QR-Rec's performance. By comparing QR-Rec against these diverse approaches, the evaluation can highlight its unique contributions and effectiveness in next item recommendation tasks.

3.7.3 Model Training

Next item recommendation is framed as a multi-class classification problem, with item ID as the class label. Binary cross-entropy is the loss function:

$$l = \sum_{u} \left(\sum_{i} -\log(\sigma(y_{i,t}^{u})) + \sum_{j} -\log(1 - \sigma(y_{j,t}^{u})) \right) \tag{3.8}$$

where σ is the sigmoid function, and i and j represent the target and negative items, respectively. A 3:1 negative sampling ratio is adopted, following [1, 2, 11].

Model parameters, including embeddings and convolutional weights, are learned from scratch. Hyperparameters such as dropout ratio and learning rate are tuned via grid search on the validation set. Computations utilize 4 Nvidia Tesla 80K GPUs and occasionally a single RTX 2080Ti GPU.

3.7.4 Evaluation Protocol

The **Leave-One-Out** approach [16, 651] is used for validation and testing. Performance metrics including **MAP, Recall@K**, and **NDCG@K** are employed:

$$MAP = \frac{\sum_{u=1}^{|U|} AP_u}{|U|} \tag{3.9}$$

$$Recall@K = \frac{|R \cap \hat{R}_{1:K}|}{|R|} \tag{3.10}$$

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{3.11}$$

A cut-off point of 10 and training for **20** epochs [16] are adopted.

3.7.5 Implementation Details

Experiments use PyTorch [656] with the Adam optimizer [631]. The learning rate is set at **0.001** after fine-tuning. A uniform batch size of **512** and an L2 regularization factor of **1e-6** are maintained across models. Dropout [657] is adjusted between **{0.2,0.5}**, and the architecture's complexity is calibrated with **[1,2,3,4]**

layers. Latent dimensions and sequence lengths are standardized at **128** and **5**, respectively. All parameters follow a normal distribution initialization.

For the Caser model [11], horizontal filters range from **[4,8,16,32,64]**, and vertical filters are tested within **[2,4,6,8]**. No additional hyperparameter tuning is required for other models. Final evaluation scores are reported consistently with the QR-Rec model.

3.8 Experiment Results and Analysis

This section delineates the results of the comprehensive experiments and the subsequent analyses. Initially, QR-Rec's performance is benchmarked against established baseline models, followed by in-depth investigations into its performance dynamics. This includes examining hyperparameters, component impacts, aggregation strategies, scale analysis, output gate contributions, and the influence of user profiles.

3.8.1 Experiment Results

In Table 3.2, QR-Rec's performance is compared against seven established baseline models across a spectrum of datasets. The most effective models for each metric are highlighted in bold, with the second-best performances underlined. This comprehensive comparison underscores QR-Rec's robustness and adaptability in various contexts:

$$\textbf{Improvement} = \textit{QR-Rec} - \textit{second best performer} \tag{3.12}$$

$$\textbf{Rel Gain} = \frac{\textbf{Improvement}}{\textit{second best performer}} \times 100\% \tag{3.13}$$

QR-Rec consistently emerges as the top performer, with absolute improvements ranging from 0.0057 to 0.0716 and relative gains between 1.44% and 17.65%. The model's effectiveness, particularly in datasets with over 99% sparsity, indicates its proficiency in handling highly sparse environments. This efficacy is attributed

TABLE 3.2: Experiment Results

Dataset	Metric	PopRec	BprMF	FMC	FPMC	Fossil	GRU4Rec	Caser	QR-Rec	Improvement	Rel Gain
Electronics	MAP	0.1739	0.3086	0.3471	0.4349	0.4287	0.2253	0.4307	**0.4846**	0.0497	11.43%
	Recall@10	0.3562	0.6046	0.5595	0.6739	0.6693	0.5150	0.6647	**0.7301**	0.0562	8.34%
	NDCG@10	0.2359	0.4332	0.4473	0.5488	0.5492	0.3208	0.5468	**0.6105**	0.0613	11.16%
Tools / Home	MAP	0.1321	0.2292	0.1576	0.2515	0.2254	0.1580	0.1227	**0.2804**	0.0289	11.49%
	Recall@10	0.2634	0.4041	0.2999	0.3948	0.4099	0.2909	0.3999	**0.4570**	0.0471	11.49%
	NDCG@10	0.1717	0.2986	0.1954	0.3032	0.2940	0.1994	0.2951	**0.3468**	0.0436	14.38%
Cell Phones	MAP	0.1456	0.3312	0.2193	0.3378	0.3207	0.1969	0.3400	**0.4000**	0.0600	17.65%
	Recall@10	0.3045	0.5369	0.4291	0.5291	0.5721	0.3902	0.5467	**0.6197**	0.0476	8.32%
	NDCG@10	0.1956	0.4182	0.2852	0.4157	0.4284	0.2603	0.4239	**0.5000**	0.0716	16.71%
Beauty	MAP	0.1309	0.3178	0.3372	0.3828	0.3512	0.2661	0.3742	**0.4025**	0.0197	5.15%
	Recall@10	0.3390	0.5618	0.4900	0.5657	0.5566	0.4715	0.5614	**0.5978**	0.0321	5.67%
	NDCG@10	0.1934	0.4176	0.4032	0.4669	0.4471	0.3479	0.4605	**0.4925**	0.0256	5.48%
Health / Care	MAP	0.1604	0.2963	0.3002	0.3638	0.3473	0.2916	0.3345	**0.3770**	0.0132	3.63%
	Recall@10	0.3713	0.5196	0.4840	0.5304	0.5485	0.4849	0.5180	**0.5741**	0.0256	4.67%
	NDCG@10	0.2193	0.3867	0.3740	0.4380	0.4361	0.3660	0.4155	**0.4662**	0.0281	6.41%
Video Games	MAP	0.1804	0.3832	0.3213	0.4027	0.3909	0.2850	0.3938	**0.4278**	0.0251	6.23%
	Recall@10	0.3794	0.6680	0.5588	0.6664	0.6664	0.5450	0.6638	**0.6949**	0.0269	4.03%
	NDCG@10	0.2447	0.5052	0.4116	0.5128	0.5091	0.3820	0.5081	**0.5445**	0.0317	6.18%
Toys / Games	MAP	0.1163	0.2796	0.2599	0.3544	0.2855	0.1968	0.3204	**0.3641**	0.0097	2.74%
	Recall@10	0.2834	0.5004	0.4279	0.5195	0.5428	0.4021	0.5205	**0.5713**	0.0285	5.25%
	NDCG@10	0.1600	0.3661	0.3259	0.4255	0.3882	0.2661	0.4070	**0.4552**	0.0297	6.98%
Sports / Outdoors	MAP	0.1570	0.2770	0.2073	0.2777	0.2475	0.1916	0.2516	**0.2872**	0.0095	3.42%
	Recall@10	0.3145	0.4782	0.3564	0.4797	0.4525	0.3748	0.4479	**0.4943**	0.0146	3.04%
	NDCG@10	0.2057	0.3545	0.2634	0.3542	0.3235	0.2502	0.3230	**0.3674**	0.0129	3.64%
Pet Supplies	MAP	0.1539	0.2843	0.2213	0.2819	0.2529	0.1956	0.2650	**0.3149**	0.0306	10.76%
	Recall@10	0.3272	0.5081	0.3947	0.4916	0.4564	0.3656	0.4616	**0.5233**	0.0152	2.99%
	NDCG@10	0.2119	0.3747	0.2815	0.3644	0.3347	0.2544	0.3422	**0.3980**	0.0233	6.22%
Digital Music	MAP	0.1237	0.2998	0.2358	0.3126	0.3169	0.2184	0.3227	**0.3683**	0.0456	14.13%
	Recall@10	0.2805	0.6009	0.4429	0.5996	0.6164	0.4513	0.5893	**0.6254**	0.0245	4.08%
	NDCG@10	0.1661	0.4238	0.3061	0.4208	0.4326	0.2982	0.4275	**0.4757**	0.0431	9.96%
Grocery / Food	MAP	0.1763	0.2542	0.3424	0.3961	0.3026	0.2733	0.3773	**0.4018**	0.0057	1.44%
	Recall@10	0.4762	0.5742	0.5148	0.5734	0.5763	0.4997	0.5647	**0.6157**	0.0394	6.84%
	NDCG@10	0.2711	0.3744	0.4210	0.4796	0.4191	0.3692	0.4651	**0.5027**	0.0231	4.82%
Instant Video	MAP	0.1421	0.2212	0.1963	0.2414	0.2107	0.1596	0.2519	**0.2825**	0.0306	12.15%
	Recall@10	0.2870	0.4463	0.3722	0.4519	0.4556	0.3426	0.4852	**0.5519**	0.0667	13.75%
	NDCG@10	0.1877	0.3085	0.2544	0.3151	0.2937	0.2169	0.3387	**0.3808**	0.0421	12.43%
Home / Kitchen	MAP	0.1565	0.3093	0.3006	0.4028	0.3448	0.2160	0.3597	**0.4148**	0.0120	2.98%
	Recall@10	0.3507	0.5741	0.4998	0.5937	0.5906	0.4117	0.5536	**0.6075**	0.0138	2.32%
	NDCG@10	0.2160	0.4216	0.3831	0.4839	0.4560	0.2897	0.4540	**0.5107**	0.0268	5.54%
Android Apps	MAP	0.2002	0.3330	0.2702	0.3670	0.3479	0.2247	0.3420	**0.4022**	0.0352	9.59%
	Recall@10	0.4099	0.6686	0.5455	0.6690	0.6623	0.4526	0.6218	**0.6808**	0.0118	1.76%
	NDCG@10	0.2648	0.4682	0.3660	0.4972	0.4820	0.2998	0.4542	**0.5212**	0.0240	4.83%
Baby	MAP	0.1458	0.1979	0.1797	0.2121	0.1955	0.1708	0.1963	**0.2322**	0.0201	9.48%
	Recall@10	0.3191	0.4120	0.3458	0.4131	0.4074	0.3632	0.3698	**0.4384**	0.0253	6.12%
	NDCG@10	0.1958	0.2761	0.2302	0.2831	0.2717	0.2321	0.2556	**0.3108**	0.0277	9.78%

to the model's ability to intricately capture local union-level features through its recurrent structure.

The analysis across various datasets reveals QR-Rec's strengths in different sectors. For instance, in the *Electronics* and *Tools/Home* categories, QR-Rec's relative gains are noteworthy, suggesting a strong capability to discern nuanced user preferences in diverse product domains. In more dynamically changing markets like *Cell Phones* and *Digital Music*, QR-Rec adapts well to the user-item interactions. In sectors such as *Beauty* and *Health/Care*, where preferences are highly personalized, QR-Rec's integration of user profiles appears to significantly enhance its recommendation precision.

The comparison with baseline models reveals distinct patterns in recommendation effectiveness. Markov Chain-based models like FPMC and Fossil often surpass

the CNN-based Caser, indicating the critical role of sequential dependencies in these datasets. The occasional success of BprMF, a non-sequential model, suggests significant non-sequential patterns in user-item interactions within certain datasets. QR-Rec's comprehensive approach, combining both user profiles and sequential data, allows it to excel beyond models focusing on singular aspects, such as GRU4Rec or FMC.

The subsequent sections will delve deeper into QR-Rec's individual components, analyzing their contributions to the model's overall success, and examine its performance in denser datasets. This in-depth analysis not only confirms QR-Rec's effectiveness across varied scenarios but also illuminates future research directions for enhancing sequential recommendation systems.

3.8.2 Key Hyperparameter Analysis

For an in-depth understanding of QR-Rec's mechanics, key hyperparameters including sequence length and latent dimension are analyzed. Two representative datasets are chosen for this exploration, and the results are illustrated in respective figures. The analysis is conducted by fixing one hyperparameter and varying the other, recording the NDCG@10 scores for each combination. The best results are highlighted in bold, and the second-best scores are underlined.

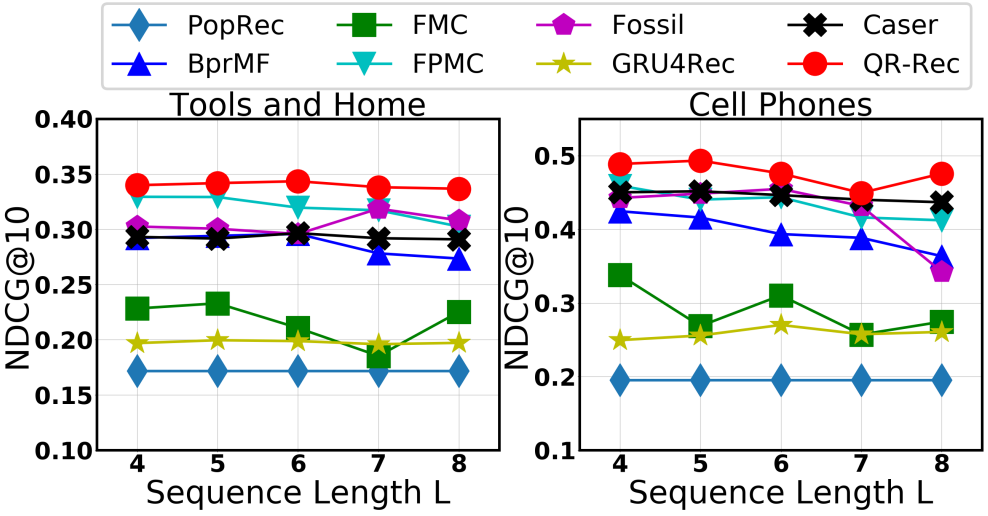


FIGURE 3.3: Anslysis of Sequence Length v.s. NDCG@10

Sequence Length. The assumption that recent items most significantly influence a user's next choice led to selecting a range of shorter sequence lengths, specifically

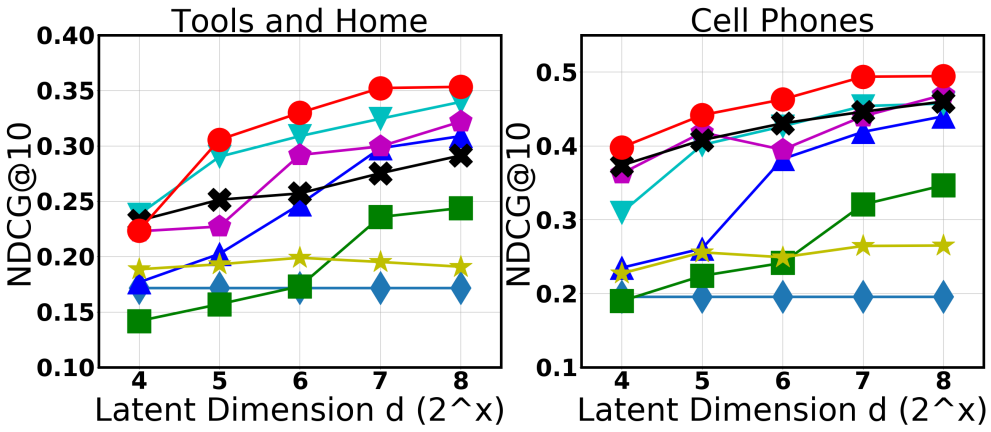


FIGURE 3.4: Analysis of Latent Dimension v.s. NDCG@10

[4,5,6,7,8], for this analysis. As illustrated in Figure 3.3, QR-Rec consistently outperforms across all tested lengths, demonstrating stability and robustness. The optimal sequence length is identified as 5, adequately capturing user behavior patterns while maintaining computational efficiency.

Latent Dimension. The latent dimension is a critical factor influencing both the evaluation outcomes and computational demands. Testing across dimensions **[16,32,64,128,256]** reveals that QR-Rec's performance consistently ranks highest. As depicted in Figure 3.4, all models exhibit a near-monotonic increase in performance with larger dimensions. However, the performance gain between 128 and 256 dimensions is marginal, indicating that 128 strikes the best balance between performance enhancement and computational resource management.

3.8.3 Aggregation Strategy Analysis

The overall architecture of QR-Rec allows for the implementation of various aggregation strategies at different levels. These strategies include using the **last hidden state** (L) of each scale's Quasi-RNN component, combined with either a **summation operation** (S) or a **mean operation** (M). Table 3.3 presents the performance of each aggregation strategy across selected datasets.

The *Hierarchical Summation Aggregation* (HSA) approach, characterized by dual summation operations, consistently yields the best results. This is evident from the marginal improvements in datasets like Tools/Home and Beauty, and more pronounced gains in Digital Music and Pet Supplies. The minor improvements

TABLE 3.3: Aggregation Strategy Analysis

Datasets	Tools / Home	Beauty	Digital Music	Pet Supplies
L+S	0.2830	0.4513	0.4082	0.3491
L+M	0.2692	0.4328	0.4397	0.3317
S+M(M+S)	<u>0.3463</u>	<u>0.4923</u>	<u>0.4558</u>	<u>0.3757</u>
M+M	0.2587	0.4244	0.4238	0.3434
HSA(S+S)	**0.3468**	**0.4925**	**0.4757**	**0.3980**
Improvement	0.0005	0.0002	0.0199	0.0223
Rel Gain	0.14%	0.04%	4.37%	5.94%

in the first two datasets suggest that scaling the results has limited impact on performance. However, the more significant gains in the latter datasets indicate a potential influence of scaling on results.

The primary advantage of HSA lies in its ability to consider and retain all features without scaling, encompassing each scale component and every unit cell. This comprehensive inclusion ensures that no valuable information is lost during aggregation, contributing to HSA's overall effectiveness compared to other strategies.

3.8.4 Scale Analysis

The multi-scale nature of QR-Rec's Quasi-RNN architecture is a pivotal aspect of its design. This section explores the efficacy of the multi-scale approach. In Table 3.4, performance metrics are displayed across four datasets, illustrating the advantages of the multi-scale setting in QR-Rec compared to single-scale Quasi-RNNs of scales 1 to 5.

The analysis reveals a significant superiority of QR-Rec over its single-scale counterparts. The best single-scale results, underlined in the table, are notably surpassed by QR-Rec, indicating the enhanced capability of the multi-scale model to capture diverse user-item interaction patterns. The fact that the most effective single scales are often 1 or 2 suggests that immediate or pairwise item relationships are predominant in these datasets. However, the inclusion of additional scales in QR-Rec leads to substantial improvements, underscoring the importance of capturing a wider range of interaction dynamics. The relative gains, notably high in

TABLE 3.4: Scale Analysis

Datasets	Tools / Home	Beauty	Digital Music	Pet Supplies
Quasi-RNN(w=1)	0.2923	0.4398	0.4400	0.3518
Quasi-RNN(w=2)	0.2659	0.4344	0.4435	0.3389
Quasi-RNN(w=3)	0.2649	0.4267	0.4344	0.3323
Quasi-RNN(w=4)	0.2644	0.4319	0.4318	0.3330
Quasi-RNN(w=5)	0.2707	0.4320	0.4346	0.3254
QR-Rec	**0.3468**	**0.4925**	**0.4757**	**0.3980**
Improvement	0.0545	0.0527	0.0199	0.0462
Rel Gain	18.65%	11.98%	7.26%	13.1%

certain datasets, reinforce the multi-scale approach's effectiveness in enhancing the model's predictive accuracy.

3.8.5 Output Gate Analysis

In the quest to optimize the RNN cell structure, a key question arises: do additional gates within a cell lead to improved performance? This inquiry is particularly relevant to QR-Rec's design, which employs dynamic average pooling with just a **forget gate**. To investigate the effectiveness of this simpler design, it is compared against a more complex variant that includes an additional **output gate**. This comparison, detailed in Table 3.5, draws upon the original Quasi-RNN structure for inspiration.

The additional output gate's integration is described by the following equations:

$$O_t^u = \sigma(W_{o,t}^u * X_t^u) \tag{3.14}$$

$$o_t^{u,N} = \sigma(W_{o,t}^{u,1} x_{t-N+1}^u + ... + W_{o,t}^{u,N-1} x_{t-1}^u + W_{o,t}^{u,N} x_t^u) \tag{3.15}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \mathbf{x}_t \tag{3.16}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t \tag{3.17}$$

These equations showcase how the output gate operates in conjunction with the cell's existing dynamics. The output gate's role is to modulate the hidden state, adding a layer of regulation to the pooling process. The comparative analysis of this setup against QR-Rec's simpler gate mechanism is encapsulated in Table 3.5.

TABLE 3.5: Output Gate Analysis

Datasets	Tools / Home	Beauty	Digital Music	Pet Supplies
with O	0.3424	0.4892	0.4514	0.3957
w/o O	**0.3468**	**0.4925**	**0.4757**	**0.3980**
Improvement	0.0044	0.0033	0.0243	0.0023
Rel Gain	1.29%	0.67%	5.38%	0.58%

The results presented in Table 3.5 reveal that QR-Rec's approach, which forgoes the output gate, typically yields better results. This suggests that in scenarios with shorter sequence lengths, like the sequence length of 5 employed in QR-Rec, adding a more complex gate may not necessarily enhance performance. Instead, it may introduce redundancy without contributing significantly to efficiency or accuracy.

3.8.6 User Profile Analysis

Incorporating user profile information (p_u) is often crucial in enhancing the performance of next item recommendation systems. The impact of integrating p_u into QR-Rec is methodically analyzed across four datasets, as depicted in Table 3.6. This analysis aims to quantify the contribution of user profile data to the overall effectiveness of the recommendation model.

TABLE 3.6: User Profile Analysis

Datasets	Tools / Home	Beauty	Digital Music	Pet Supplies
p_u only (**1**)	0.2923	0.4002	0.4390	0.3678
QR-Rec w/o p_u (**2**)	0.3382	0.4879	0.4450	0.3570
QR-Rec	**0.3468**	**0.4925**	**0.4757**	**0.3980**
QR-Rec - (**1**)	0.0545	0.0923	0.0367	0.0302
Rel Gain	18.65%	23.06%	8.36%	8.21%
QR-Rec - (**2**)	0.0086	0.0046	0.0307	0.0410
Rel Gain	2.54%	11.98%	6.90%	11.48%

The table elucidates the enhancements brought by the inclusion of user profile data. The first comparison, QR-Rec - (**1**), reveals significant improvements across all datasets when the core module of QR-Rec is augmented with p_u. These substantial

gains, ranging from 8.21% to 23.06%, underscore the effectiveness of QR-Rec's core module in leveraging user profile data.

Further analysis, QR-Rec **-** (**2**), compares the performance of QR-Rec with and without user profile integration. Here again, the results affirm the value of incorporating p_u, with relative gains varying from 2.54% to 11.48%. This improvement, though smaller than the first comparison, still highlights the significant role user profiles play in refining the recommendation process.

Overall, the results presented in Table 3.6 substantiate the substantial impact of user profile data on enhancing the predictive accuracy of QR-Rec. This confirms the model's proficiency in not only capturing item interaction dynamics but also effectively utilizing user-specific information to deliver more personalized and relevant recommendations.

3.9 Summary

This chapter introduces **QR-Rec**, a novel neural inductive bias specifically designed for next-item recommendation tasks. By addressing the first research question posed in Section 1.2—*How to model item order information beyond a single fixed-length user interaction sequence for next-item recommendation?*—QR-Rec leverages a multi-scale *Quasi-Recurrent Neural Network* (Quasi-RNN) architecture enhanced with dynamic average pooling caping with the gap of fixed-length single-scale user history. Convolutional filters of various sizes efficiently extract an extensive range of local and union-level features from user-item interaction sequences, and these features are integrated into the recurrent structure to capture the most comprehensive user-item interaction features. Another key innovation of QR-Rec lies in its *Hierarchical Summation Aggregation* strategy, which aggregates outputs from each architectural level by summation, preserving richer information and improving recommendation accuracy. The incorporation of user profile information further refines personalized predictions by aligning learned representations with individual user preferences. Across extensive experiments on 15 diverse and highly sparse Amazon Reviews datasets, QR-Rec demonstrates significant performance gains over competing methods, underscoring the robustness and adaptability of its multi-scale design. Collectively, the proposed QR-Rec model provides

a more comprehensive treatment of sequential interactions by effectively unifying CNN-derived features and RNN-based temporal modeling. This approach not only enhances next-item recommendation precision but also offers a versatile framework for incorporating additional user-specific information in future research.

Chapter 4

Transitional Graph Attentional Net (TransGAT)

4.1 Introduction

In the realm of recommendation systems, the concept of next-basket recommendation emerges as a notably complex scenario, surpassing the simplicity of next-item recommendation. This complexity stems from intricate data structures that demand not only the understanding of multiple items' transitional relations across baskets but also the prediction of entire baskets, rather than individual items. To address these multifaceted challenges, a pioneering model known as ***Trans****itional* ***G****raph* ***AT****tention Net* (**TransGAT**) is introduced. TransGAT employs the Graph Attention Network (GAT), an advanced graph neural network, which is particularly effective in scenarios where the significance of relationships within graph structures plays a crucial role. TransGAT excels in capturing and modeling the dynamic relations between consecutive baskets. Furthermore, the integration of a self-attention mechanism within this graph network framework enhances TransGAT's ability to discern the relative importance of these inter-basket relationships. This advanced approach positions TransGAT at the forefront of significant improvements in basket-level recommendation systems, adeptly handling the intricate requirements of next-basket scenarios with innovative and insightful solutions.

[2]The work in this chapter has been submitted as "Transitional Graph Attention Network for Next Basket Recommendation" to the International Journal of Crowd Science.

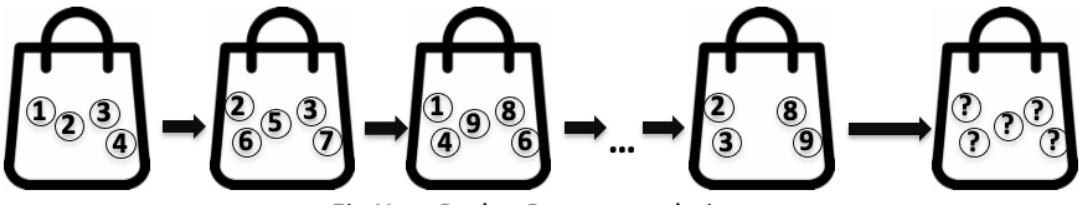


FIGURE 4.1: Next-Basket Recommendation Illustration

4.2 Problem Definitions and Formulations

Next basket recommendation (Figure 4.1) is aptly formalized as a sequential prediction problem, which is delineated in the context of user behaviors and item interactions. Consider $U = \{u_1, u_2, \ldots, u_{|U|}\}$ as the set representing users, and $I = \{i_1, i_2, \ldots, i_{|I|}\}$ as the set denoting items. Each user u in U is associated with a distinct sequence of purchase behaviors, denoted by $S^u = (B^u_{t1}, B^u_{t2}, \ldots, B^u_{t_{L-1}})$. Here, B^u_t represents the basket of items purchased by user u at the time step t. The core objective of this sequential prediction problem is to accurately forecast the composition of the next basket $B^u_{t_L}$ for each user u, given their observed purchasing sequence S^u. This formulation encapsulates the challenge of understanding and predicting user preferences over time, a task fundamental to the efficacy of recommendation systems in anticipating future needs.

The prediction of the next basket can be mathematically formulated as:

$$\hat{B}^u_{t_L} = f(S^u; \Theta) \tag{4.1}$$

where $\hat{B}^u_{t_L}$ is the predicted basket for user u at time t_L, S^u is the observed sequence of past baskets, and $f(\cdot\,; \Theta)$ represents the predictive model parameterized by Θ. The goal is to optimize Θ such that the predictive model accurately anticipates the composition of the next basket, reflecting the evolving preferences and behaviors of the user.

Example in Grocery Shopping: Consider a grocery shopping scenario where user u has the following sequence of baskets:

$$S^u = (\{\text{milk, bread}\}, \{\text{eggs, cheese}\}, \{\text{chicken, rice}\})$$

The task is to predict the next basket $\hat{B}^u_{t_L}$ that the user is likely to purchase. Based on the user's past baskets, the model might predict

$$\hat{B}^u_{t_L} = \{\text{beef, pasta}\}$$

This prediction helps in providing personalized shopping recommendations to the user, enhancing their shopping experience.

Example in Online Retail: In an online retail platform, a user u has purchased the following sequence of item baskets:

$$S^u = (\{\text{laptop, mouse}\}, \{\text{keyboard, monitor}\}, \{\text{printer, paper}\})$$

The task is to predict the next basket $\hat{B}^u_{t_L}$ the user is likely to purchase. Based on this sequence, the model might predict

$$\hat{B}^u_{t_L} = \{\text{USB drive, external hard drive}\}$$

This recommendation can help in targeting the user's future needs more accurately.

Example in Fashion Retail: On a fashion retail platform, a user u has bought the following sequence of baskets:

$$S^u = (\{\text{t-shirt, jeans}\}, \{\text{jacket, sneakers}\}, \{\text{dress, heels}\})$$

The task is to predict the next basket $\hat{B}^u_{t_L}$ the user is likely to purchase. Given this history, the model might predict

$$\hat{B}^u_{t_L} = \{\text{sunglasses, handbag}\}$$

Accurate prediction of the next basket helps in recommending complementary fashion items to the user, thereby increasing sales and customer satisfaction.

4.3 Existing Shortcomings

In the realm of next-basket recommendation, numerous approaches have been explored, each with distinct limitations. The Markov chain-based method FPMC [1]

combines the strengths of Markov chains and matrix factorization but fails to adequately model interactions beyond immediate successive items. Neural methods, while offering advancements, also face challenges: NNRec [27] relies on Multilayer Perceptrons (MLPs) without addressing complex transition modeling, and HRM [26] employs basic aggregation and concatenation, neglecting crucial basket transitions. DREAM [7] represents a significant step forward by utilizing Recurrent Neural Networks (RNNs) to model basket sequences, yet it overlooks the relationships among items across consecutive baskets. ANAM [29] integrates an attention mechanism with RNNs, enhancing performance but still falling short in capturing the finer, granular features essential for understanding items' transitional dynamics across baskets. These shortcomings highlight the necessity for a more holistic and nuanced model capable of deciphering the complex patterns inherent in next-basket recommendation scenarios. None of the aforementioned methods attempt to model the transitional relationships between items across baskets at the time when this work was conducted, which is one of the main focuses of this thesis introduced in Section 1.2.

4.4 Contributions

This chapter presents several significant contributions to the field of next-basket recommendation, specifically addressing the second research question outlined in Section 1.2: *How to represent basket items relationships in the transitional sequences effectively for next basket recommendation?*

- This study introduces a novel Transitional Graph Attention Network (TransGAT) architecture that pioneers the application of Graph Attention Networks (GAT) to next-basket recommendation tasks. The architecture constructs a transitional graph to represent the relational dynamics of items across baskets, where a shared self-attention mechanism updates each item's node embedding by considering the weighted influence of neighboring nodes. By effectively capturing the complex transitional relationships between items across consecutive baskets through the integration of graph structures and self-attention mechanisms, this approach enhances the understanding of user-item interactions and enriches the model's ability to represent and learn transitional item relationships, leading to improved recommendation accuracy.

- The study introduces a novel basket representation approach that employs average pooling to create unified embeddings. By aggregating item embeddings within each basket through average pooling, this method generates a comprehensive basket-level representation that captures the collective characteristics of all items in the basket. This unified embedding approach enables effective modeling of basket-level patterns and transitions, facilitating more accurate predictions of future basket contents.

- Extensive experiments were conducted on various real-world datasets from major e-commerce platforms, including TaFeng, T-Mall, and Amazon. The results demonstrate that the proposed model outperforms existing state-of-the-art methods in key metrics such as Hit Ratios and Normalized Discounted Cumulative Gains (NDCGs) at various cutoffs. Additionally, an ablation study and a series of analytical evaluations provide deeper insights into the factors contributing to the model's enhanced performance, validating the effectiveness of the approach.

The findings of this research signify a notable advancement in next-basket recommendation. They underscore the potential of graph-neural-network-based approaches in addressing the complexities of these recommendation scenarios.

4.5 Related Work

This section reviews the existing literature in the fields of graph neural networks, graph attention networks, and next basket recommendation. The focus is on advancements and applications that have significantly contributed to the development of effective recommendation systems.

4.5.1 Graph Attention Networks

Graph attention networks (GATs), such as those introduced by [602], represent a significant advancement in graph neural networks, operating on graph-structured data with masked self-attentional layers for focused processing of neighboring nodes' features. In the recommendation systems domain, graph-based methods

like KGAT [658] have been applied to knowledge graphs, integrating side information for enhanced recommendations. The Dual graph attention networks [5] address social recommendation by capturing dual social effects through user-specific and dynamic context-aware attention weights. Additionally, [71] proposed a dynamic graph attention network that combines RNNs for dynamic user behavior analysis with graph attention networks for real-time influencer identification. TransGAT stands out as a pioneering method in using graph attention networks for modeling item transition relations in next basket recommendation tasks.

4.5.2 Next Basket Recommendation

The next basket recommendation focuses on predicting the contents of a user's future purchases, utilizing various innovative approaches. FPMC[1] combines matrix factorization with Markov chains to model personalized sequences of user interactions effectively. NN-Rec[27] leverages neural networks to create dynamic representations of user preferences, adapting to changes over time. HRM[26] employs hierarchical representation learning to capture multi-level features of user behavior. DREAM[7] integrates dynamic recurrent networks to model temporal dynamics in user purchase patterns. ANAM (NAM)[29] focuses on attribute-aware models, enhancing recommendations by considering item attributes. Triple2vec-adaLoyal[67] introduces loyalty modeling to account for user loyalty in prediction. SecGT[369] incorporates graph neural networks to model complex relationships between items. CCBT[377] utilizes contextual information to improve recommendation accuracy. BFM/CBFM[84] and BSEQ[89] offer models specifically designed for basket sequences, capturing the order and grouping of purchases. Beacon[96] leverages correlation networks to find associations between items. BRL[397] employs reinforcement learning to optimize long-term user satisfaction. Basket random walk[99] uses random walk techniques to model user navigation through item space. Neural-basket-emb[255] focuses on creating neural embeddings for items, enhancing the model's understanding of item similarities. Hyper-conv-NBRR[259] uses hypergraph convolution to capture high-order interactions among items. GenRec[409] integrates generative models to predict user behavior under different scenarios. TAIW[414] focuses on time-aware models, incorporating temporal dynamics into predictions. BTBR[422] utilizes masked transformers to model sequential data effectively. TIFUKNN[154] employs k-nearest neighbors to predict next purchases

based on similar users' behavior. CLEA[281] integrates world knowledge into the recommendation process, enhancing the model's understanding of item contexts. ReCANet[287] uses attention networks to focus on relevant parts of the user's purchase history. MMNR[430] employs multi-modal learning to integrate various data sources for improved recommendations. DigBot[435] leverages dialogue systems to interact with users and refine recommendations based on real-time feedback. TREx[448] integrates explainability, allowing users to understand the reasoning behind recommendations. M2[334] utilizes meta-learning to adapt to new users and items quickly. HapCL[515] employs hierarchical contrastive learning to improve the differentiation between user preferences. HAEM[335] focuses on interactive attention mechanisms to refine user models based on interactions. NBR-reality-check[527] validates model robustness, ensuring the reliability of predictions. These advancements collectively enhance the ability to predict users' next purchases accurately, taking into account various factors such as temporal dynamics, item attributes, user loyalty, and contextual information.

4.5.3 GNN-based Sequential Recommendation

Graph neural networks (GNNs) have been widely applied in sequential recommendation tasks, leveraging the inherent graph structures in user-item interactions. Methods like SRGNN [21] and STAMP [14] have demonstrated the effectiveness of GNNs in capturing sequential patterns and user preferences. Building on this foundation, TAGNN [24] upgrades SRGNN by incorporating GGNN and a target attention mechanism to calculate scores for all items in a session with respect to the target. Similarly, DGRec [71] uses a dynamic-graph-attention neural network to model context-dependent social influence, dynamically inferring influencers based on users' current interests. FGNN [22] addresses the next item recommendation task through graph classification, employing a weighted attention graph layer and a Readout function to derive embeddings for items and sessions. GAG [150] further extends this by transforming the session into a weighted graph with global attributes, using graph convolution on node features, edge features, and global attributes. HyperRec [138] leverages a hypergraph to model item correlations, connecting multiple items with one hyperedge and utilizing hypergraph convolution to capture both direct and high-order connections as short-term correlations between items. MAGNN [23] integrates GNN for short-term interest modeling with

a memory net for long-term interest modeling and a bi-linear function for item co-occurrence patterns. GCE-GNN [25] combines GCN and GAT layers for global and session-level item representations, while GLSGRL [137] uses GNNs to represent users' long and short-term preferences. MKMSR [139] introduces micro-behavior sequences to learn embeddings via GGNN and GRU. DGNN [474] decouples the modeling of explicit dependencies and implicit correlations among items using a single gate GNN (SG-GNN) and an adaptive GNN (A-GNN). RetaGNN [313] introduces relational attentive weight matrices in the enclosing subgraphs centered at the target user-item pair without relying on content and auxiliary information. HG-GNN [307] constructs a heterogeneous global graph considering historical user interactions, item transitions, and global co-occurrence information. Lastly, SERec [300] constructs a heterogeneous knowledge graph from social networks and historical user behaviors, applying a heterogeneous GNN to learn KG embeddings of users and items, which are then fed into an SR model to enhance recommendation accuracy. The continuous evolution and incorporation of various GNN-based techniques have significantly improved the accuracy and relevance of recommendations, underscoring the importance of leveraging graph structures in modeling user behaviors.

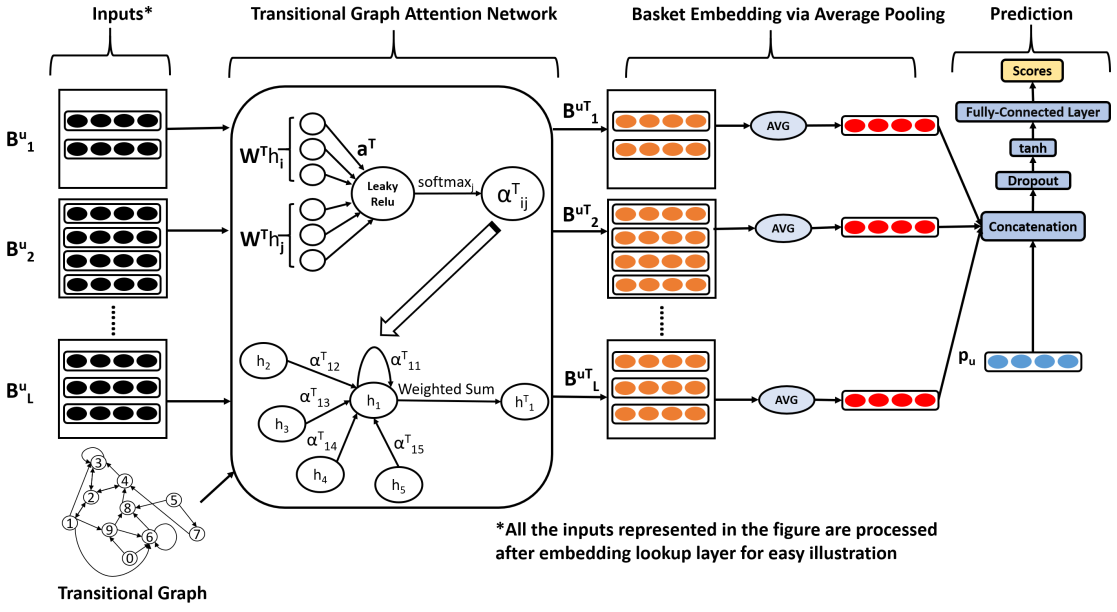


FIGURE 4.2: The Transitional Graph Attention Net (TransGAT) Architecture

4.6 Methodology

The proposed methodology for next-basket recommendation integrates embedding techniques, graph attention networks, and predictive modeling into a cohesive framework. The initial phase involves an Embedding Lookup Layer that maps users and items into a high-dimensional latent space. At the core of this approach is the Transitional Graph Attention Layer, where a graph constructed from items' co-occurrences in baskets undergoes analysis via a graph attention network. This network discerns the relative importance of items, thereby capturing their dynamic relationships. Basket embeddings are subsequently created by averaging these attention-informed item embeddings, effectively summarizing the basket's collective characteristics. In the final stage, a Prediction Layer processes these basket embeddings, in conjunction with user profiles, through neural layers to predict the contents of the subsequent basket. This methodological design aims to capture the complex, sequential interplay of user-item interactions, offering an insightful understanding of their underlying dynamics.

Figure 4.2 illustrates the overall architecture of the model:

1. Inputs to the transitional graph attention network are item embeddings (depicted in black) within each basket and the transitional graph.
2. Inside the transitional graph attention network, attention coefficients are computed following a linear transformation and a non-linear operation, resulting in weighted summation outputs.
3. The updated item embeddings (shown in orange) are then aggregated through average pooling to obtain basket embeddings (represented in red).
4. Finally, these embeddings are concatenated with the user embedding (indicated in blue) to generate predictive scores.

4.6.1 Embedding Lookup Layer

The Embedding Lookup Layer is a crucial component that maps identifiers (IDs) to vectors in a specified dimensional space. This process is essential for transforming raw data into a format suitable for subsequent neural network processing. In the

context of this model, every item in a basket, as part of a sequence of length L, is mapped into a latent space of dimension d. Mathematically, this can be represented as follows:

$$h_i = \text{Embedding}(i; \mathcal{E}_I) \quad \forall i \in \mathbb{I} \tag{4.2}$$

where $h_i \in \mathbb{R}^d$ denotes the embedding of item i, and $\mathcal{E}_I$ represents the embedding matrix for items. Similarly, each user is mapped into the same latent space, providing a dense representation of their profile:

$$p_u = \text{Embedding}(u; \mathcal{E}_U) \quad \forall u \in \mathbb{U} \tag{4.3}$$

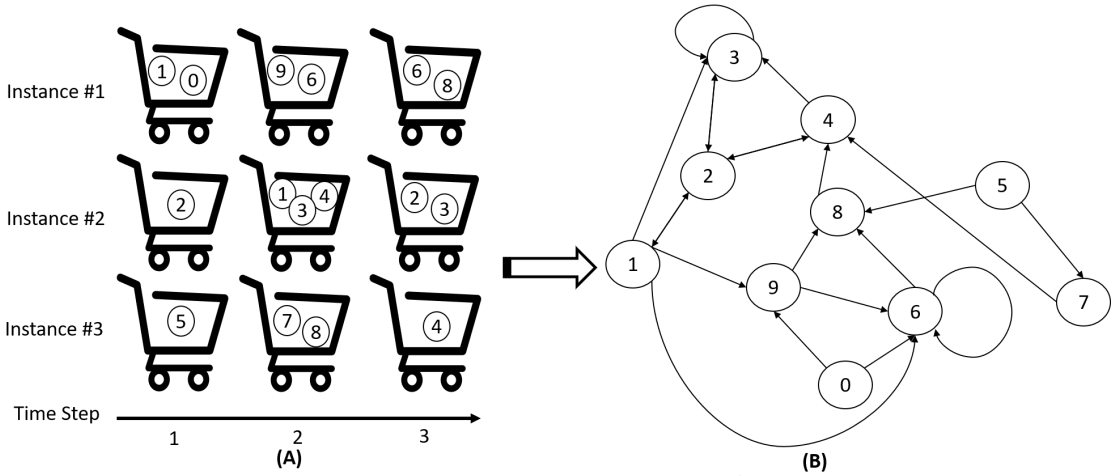
where $p_u \in \mathbb{R}^d$ is the user embedding for user u, and $\mathcal{E}_U$ is the embedding matrix for users. This layer effectively encodes both items and users into a high-dimensional space where their relationships can be more readily discerned and utilized in the recommendation process.

4.6.2 Transitional Graph Attention Layer

The Transitional Graph Attention Layer is an innovative aspect of this model, addressing two crucial tasks: constructing the transitional graph and applying the graph attention network. The inputs include item embeddings $h_i \in \mathbb{R}^d$ within each basket and embeddings $h_j \in \mathbb{R}^d$ of items in consecutive baskets, along with a pre-constructed transitional graph. The output is the transformed embeddings $h_i^T \in \mathbb{R}^d$ for $i, j \in \mathbb{I}$.

Transitional Graph Building. The transitional graph is constructed by representing each item as a node. Directed edges are added between nodes if the corresponding items occur in consecutive baskets in the training data. This results in a graph where the nodes are interconnected based on their sequential appearance, encapsulating the transitional dynamics of item interactions. The graph is efficiently represented as a sparse adjacency matrix, where an entry ij is set to 1 if item j is a neighbor of item i (i.e., they appear in consecutive baskets), and 0 otherwise. Figure 4.3 illustrates this graph construction process with an example. Sub-figure (A) shows sample user-basket sequences, sub-figure (B) depicts the

resulting transitional graph, and sub-figure (C) represents the graph as a sparse matrix.



	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	1	0	0	1
1	0	0	1	1	0	0	1	0	0	1
2	0	1	0	1	1	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0
4	0	0	1	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	1	0
6	0	0	0	0	0	0	1	0	1	0
7	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	1	0	1	9

(C)

FIGURE 4.3: Graph and Matrix Representations Example

Transitional Graph Attention Network. In this network, item embeddings from the embedding layer serve as nodes. These nodes, represented as h_i, are coupled with their transitional neighbors h_j as inputs. The embeddings are first linearly transformed to enhance their expressive power:

$$\tilde{h}_i = W^T h_i \quad \forall i \in \mathbb{I} \tag{4.4}$$

where $W^T \in \mathbb{R}^{d \times d}$ is the linear transformation matrix. The self-attention mechanism, parameterized by a weight vector $a^T \in \mathbb{R}^{2d}$, then computes attention coefficients e_{ij}^T:

$$e_{ij}^T = \text{LeakyReLU}\left(a^T \left[\tilde{h}_i \parallel \tilde{h}_j\right]\right) \tag{4.5}$$

These coefficients, indicative of the importance of item j to item i in the transitional context, are normalized using the softmax function to ensure comparability:

$$\alpha_{ij}^T = \frac{\exp(e_{ij}^T)}{\sum_{k \in N_i^T} \exp(e_{ik}^T)} \tag{4.6}$$

The attention mechanism's output is used to compute the final node embeddings, capturing the relational context:

$$h_i^T = \sigma\left(\sum_{j \in N_i} \alpha_{ij}^T \tilde{h}_j\right) \tag{4.7}$$

where σ is an activation function, here chosen as *Elu*.

Basket Embedding. To represent each basket as a single embedding, an average pooling operation is employed. This step aggregates the individual item embeddings, processed through the graph attention network, into a cohesive basket-level representation:

$$h_t^B = \frac{1}{|B_t^u|} \sum_{i \in B_t^u} h_i^T \tag{4.8}$$

where h_t^B is the embedding of basket B_t^u, and $|B_t^u|$ denotes the number of items in the basket. This aggregated basket embedding captures the collective characteristics of the items within each basket, crucial for the subsequent prediction task.

4.6.3 Prediction Layer and Recommendation

The final step in the TransGAT model is the prediction layer, which synthesizes the processed basket embeddings to predict the composition of the user's next basket. This layer utilizes a concatenation of basket embeddings, followed by a series of

neural transformations to generate predictive scores for potential items in the next basket.

Concatenation of Basket Embeddings. The basket embeddings, obtained from the Transitional Graph Attention Layer, encapsulate the collective characteristics of items within each basket. These embeddings are concatenated to form a comprehensive representation of the user's sequential basket history:

$$o_{t+1}^u = [h_1^B \parallel h_2^B \parallel \ldots \parallel h_t^B] \tag{4.9}$$

where o_{t+1}^u denotes the concatenated basket embeddings for user u up to time t.

Integration with User Profile. To incorporate the general preferences of user u, represented by the user embedding p_u, this embedding is concatenated with o_{t+1}^u. The concatenated vector is then processed through a dropout layer, followed by a **tanh** activation function, and a fully-connected layer to generate the final predictive scores for all items:

$$y_{t+1}^u = W^u \cdot \tanh(\text{dropout}([o_{t+1}^u \parallel p_u])) + b^u \tag{4.10}$$

Here, $W^u \in \mathbb{R}^{2d \times |I|}$ and $b^u \in \mathbb{R}^{|I|}$ are learnable parameters, with $|I|$ being the total number of items.

Recommendation Process. Upon completion of training, the user's embedding and their recent basket embeddings serve as inputs to the model. The top N items with the highest scores in y_{t+1}^u are recommended, reflecting the model's prediction for the user's next basket. Unlike traditional single-label classification problems where softmax is commonly used, next-basket recommendation inherently involves multiple labels. Therefore, the model is designed to handle this multi-label multi-class classification, allowing for the prediction of a basket comprising multiple items. This approach ensures a tailored and accurate recommendation for each user, aligning with their unique purchasing history and preferences.

4.7 Experimental Setup

This section describes the datasets, baselines, and evaluation metrics used in the experiments. The implementation details are also presented.

TABLE 4.1: Statistics of the datasets

Dataset	#users	#items	#baskets	#interactions	sparsity	#categories	max basket size	avg #items per basket
TaFeng	4419	10817	54897	714893	98.50%	1357	109	13.02
T-Mall	5257	13202	80294	143058	99.79%	70	20	1.78
Android Apps	4871	7191	42998	86989	99.75%	Nil	58	2.02

4.7.1 Datasets

This study employs datasets from three distinct sources: **TaFeng**, **T-Mall**, and **Android Apps** from Amazon Reviews, each offering a unique perspective on consumer purchasing behaviors.

TaFeng Dataset [1] encompasses four months of comprehensive basket purchase data from a grocery store, providing insights into consumer buying patterns in a supermarket context. It is suitable for next basket recommendation as it captures frequent and varied purchases. Each day's purchases can be aggregated into a basket, forming a sequence of baskets over time.

T-Mall Dataset [2], released by the Alibaba Group, this dataset captures the vibrancy of e-commerce interactions in China, reflecting a diverse range of consumer choices on a large online shopping platform. Its large scale and variety make it an excellent source for studying next basket recommendations in an online retail setting. Purchases made on the same day can be aggregated into a basket to form sequences of baskets.

Amazon Reviews Dataset [3], specifically the **Android Apps** section, offers a deep dive into user-item interactions on Amazon's platform. Although the primary focus is on the interactions rather than the review content, it provides valuable insights into user preferences and behaviors. [651–653]. It is suitable for next basket recommendation studies where users often purchase multiple related items on the same day.

[1] `http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng`
[2] `https://tianchi.aliyun.com/dataset/dataDetail?dataId=53&userId=1`
[3] `http://jmcauley.ucsd.edu/data/amazon/`

Preprocessing Steps: The datasets undergo several preprocessing procedures to ensure consistency and relevance:

- Timestamp features are utilized to establish the chronological order of user interactions within their historical sequences.
- Users with fewer than **10** interactions and items with fewer than **10** users are filtered out [1, 11, 651, 652, 654, 655]. Additionally, users with less than **7** baskets are discarded after aggregating interactions into basket sequences. This step aims to maintain comparably long sequences for each user, focusing primarily on warm-start scenarios and reducing cold-start noise.
- The Amazon dataset is processed to include only user-item interactions rated **3** or higher, indicating a preference. Given the absence of explicit baskets in the Amazon data, items reviewed within a single day are aggregated to form a basket, capturing daily purchasing patterns.

Table 4.1 presents the key statistics of these datasets, offering a comprehensive overview of the data employed in this study.

4.7.2 Baselines

TransGAT is compared against **8** diverse and well-established baselines in the field of recommendation systems at the time this research was conducted, from popularity-based methods to advanced neural network models. These baselines are chosen to represent a broad spectrum of recommendation techniques, enabling a comprehensive evaluation of the proposed model's performance. The baselines are as follows:

- **TOP**: This method recommends items based on their overall popularity in the dataset, quantified by item frequency. The recommendation of Top-N items for any user is determined by the frequency of items in the dataset.
- **BprMF** [576]: Utilizes Bayesian Personalized Ranking with Matrix Factorization for non-sequential recommendation scenarios. For next-basket recommendation, it predicts the next basket by leveraging user-item interaction

data thourgh matrix factorization via the objective function of Bayesian Personalized Ranking.

- **FMC**: Utilizes Matrix Factorization on a first-order Markov transition matrix for sequential recommendation. It models the transition probability between items in consecutive baskets using matrix factorization.

- **FPMC** [1]: Combines the strengths of Markov Chains and Matrix Factorization to model user-item interactions. It factorizes the transition matrix of items in consecutive baskets to predict the next basket.

- **HRM** [26]: Implements a hierarchical representation model using neural networks for integrating user and item purchase history. It models user behavior by learning hierarchical representations of user-item interactions. To predict the next basket, it concatenates user and item embeddings to capture sequential dependencies utilizing neural networks of the hierarchical representation model.

- **DREAM** [7]: Incorporates RNNs to model both user preferences and sequential patterns in purchase history. It uses dynamic RNNs to capture temporal dynamics in user-item interactions. For next basket recommendation, it predicts the next basket based on the user's historical basket sequences via RNNs.

- **ANAM (NAM)** [29]: Deploys an attentive RNN mechanism to capture temporal user behavior, differentiating between ANAM and NAM based on the availability of category data. It uses an attention mechanism with RNNs to model user-item interactions, focusing on item attributes. For next basket recommendation, it predicts the next basket based on the user's historical basket sequences via an attention mechanism.

- **NN-Rec** [27]: Uses item embedding and aggregation techniques, coupled with a nonlinear activation function, to model sequential dependencies. It employs MLP to capture user-item interactions and predict the next basket of items based on historical basket sequences.

4.7.3 Model Training

The objective function for model training is the binary cross-entropy loss, applied to a multi-label classification setting. The loss function is defined as:

$$l = \sum_{u}(\sum_{i} -\log(\sigma(y^{u}_{i,t})) + \sum_{j} -log(1 - \sigma(y^{u}_{j,t}))) \tag{4.11}$$

where σ denotes the sigmoid function, and i and j represent target and negative items, respectively. A 3:1 negative sampling ratio is adopted, following [1, 2, 11].

Hyperparameters, including dropout ratio, learning rate, and L2 regularization, are fine-tuned using grid search on the validation set.

4.7.4 Evaluation Protocol

The Leave-One-Out evaluation method [16, 651] is utilized, where for each user, the last basket is held out for testing, and the remaining baskets are used for training. The model predicts the contents of the held-out basket, and the performance is evaluated based on the accuracy of these predictions.

The evaluation metrics used in this study are:

- **Hit Ratio@K (HR@K)**: Measures whether the target items are present in the top-K recommendation list.
- **NDCG@K**: Evaluates the quality of ranking by assigning higher weights to items ranked higher.

The evaluation employs cutoff points at 10 and 20, in alignment with standard practices.

4.7.5 Implementation Details

Implemented in PyTorch, the models are optimized using either Adam or SGD. Specific parameters such as learning rates, L2 regularization rates, and latent dimensions are tailored to each dataset. Batch size is set to 64 for all models, and

dropout rates are adjusted between {0.2, 0.5}. The models undergo training for 30 epochs, and the final evaluation is based on the best performing settings during validation.

4.8 Experiment Results and Analysis

This section presents the several experiment results and analysis, which is composed of experiment results, ablation study, sequence length analysis and dimention analysis subsections.

TABLE 4.2: Experiment Results

Dataset	Metric	TOP	BprMF	FMC	FPMC	HRM	DREAM	ANAM(NAM)	NNRec	TransGAT	Relative Gain
TaFeng	HR@10	0.3136	0.3241	0.2725	0.2835	0.2989	0.3243	0.3200	0.3419	**0.3503**	2.46%
	HR@20	0.3607	0.4003	0.3661	0.3779	0.3607	0.3788	0.3591	0.4248	**0.4331**	1.95%
	NDCG@10	0.2350	0.2412	0.2219	0.2271	0.2331	0.2383	0.2387	0.2501	**0.2538**	1.48%
	NDCG@20	0.2410	0.2518	0.2345	0.2419	0.2401	0.2443	0.2436	0.2618	**0.2635**	0.65%
T-Mall	HR@10	0.0726	0.0858	0.0711	0.0772	0.0824	0.0710	0.0713	0.0926	**0.0964**	4.10%
	HR@20	0.0788	0.1006	0.0795	0.0890	0.0945	0.0799	0.0797	0.1117	**0.1187**	6.27%
	NDCG@10	0.0668	0.0728	0.0647	0.0686	0.0718	0.0647	0.0654	0.0787	**0.0793**	0.76%
	NDCG@20	0.0681	0.0761	0.0665	0.0713	0.0743	0.0668	0.0673	0.0830	**0.0843**	1.57%
Android Apps	HR@10	0.0482	0.0636	0.0466	0.0505	0.0509	0.0585	0.0585	0.0661	**0.0710**	7.41%
	HR@20	0.0593	0.0864	0.0693	0.0630	0.0647	0.0918	0.0932	0.0961	**0.1100**	14.46%
	NDCG@10	0.0279	0.0373	0.0295	0.0307	0.0308	0.0359	0.0349	0.0430	**0.0440**	2.33%
	NDCG@20	0.0308	0.0428	0.0338	0.0339	0.0343	0.0445	0.0436	0.0504	**0.0540**	7.14%

4.8.1 Experiment Results and Analysis

Table 4.2 details the comparative performance of TransGAT against eight baseline models across three different datasets: TaFeng, T-Mall, and Android Apps. The metrics employed for evaluation include Hit Ratio (HR) at cutoffs of 10 and 20, and Normalized Discounted Cumulative Gain (NDCG) at the same cutoffs. The relative gain for TransGAT over the second-best performer is calculated as follows:

$$\textbf{Relative Gain} = \frac{\textit{TransGAT} - \textit{Second Best Performer}}{\textit{Second Best Performer}} \times 100\% \tag{4.12}$$

TransGAT consistently outperforms all baseline models in both HR and NDCG across all datasets. Notably, the relative gain in HR@10 ranges from 2.46% to 7.41%, and for HR@20, it varies from 1.95% to an impressive 14.46%. This performance trend is indicative of TransGAT's effectiveness in harnessing graph-structured features from items' transitional relations across baskets.

Interestingly, the NNRec model exhibits strong performance, highlighting the efficacy of simple architectures that leverage item embeddings and mean aggregation. This aligns with the principle of Occam's Razor, suggesting that simplicity can be effective in certain contexts. Despite its simplicity, NNRec outperforms several other baselines, validating the significance of capturing recurrent item importance in basket recommendations.

Furthermore, models like FPMC, which include user profile information, demonstrate superior performance compared to those that do not, emphasizing the importance of personalization in recommendation systems. The Top model, while generally outperformed by other baselines, still shows competitive results in some metrics, particularly on the TaFeng and T-Mall datasets. This observation reinforces the notion that popular items can sometimes be effective recommendations, though they are unable to match the comprehensive performance of TransGAT.

In summary, the results from these experiments highlight the superiority of TransGAT in next-basket recommendation tasks, showcasing its ability to effectively capture and utilize the complex relational dynamics present in user-item interaction data.

4.8.2 Ablation Study

The ablation study for TransGAT, as outlined in Table 4.3, offers insightful observations on the effectiveness of the transitional graph attention network (GAT) layer. This study compares the performance of TransGAT with and without the GAT layer across two datasets, TaFeng and Android Apps.

Table 4.3: Ablation Study

Dataset	Metric	Without GAT	TransGAT	Relative Gain
TaFeng	HR@10	0.3469	**0.3503**	0.98%
	HR@20	0.4288	**0.4331**	1.00%
	NDCG@10	0.2480	**0.2538**	2.34%
	NDCG@20	0.2597	**0.2635**	1.46%
Android Apps	HR@10	0.0671	**0.0710**	5.81%
	HR@20	0.1047	**0.1100**	5.06%
	NDCG@10	0.0381	**0.0440**	15.49%
	NDCG@20	0.0477	**0.0540**	13.21%

The results from the TaFeng dataset reveal a noticeable improvement when the GAT layer is included. Both Hit Ratios (HR@10 and HR@20) and Normalized Discounted Cumulative Gains (NDCG@10 and NDCG@20) show enhanced performance with the GAT layer, underscoring its contribution to accurately capturing the transitional dynamics between items across baskets.

In the Android Apps dataset, the impact of the GAT layer is even more significant. The model demonstrates substantial enhancements in all metrics. The improvements in HR@10 and HR@20, as well as the considerable gains in both NDCG metrics, highlight the GAT layer's ability to effectively model complex item relationships in the context of next-basket recommendations.

These findings validate the importance of the transitional graph attention network in the TransGAT model. Its inclusion not only enhances the model's ability to process complex relational information but also significantly boosts its predictive performance, showcasing its potential in improving next-basket recommendation tasks.

4.8.3 Sequence Length Analysis

An analysis focusing on the impact of sequence length on the performance of TransGAT was conducted using the TaFeng and Android Apps datasets. The analysis, detailed in Table 4.4, explores how varying the sequence length (denoted as L) influences the model's effectiveness in predicting the next basket.

For the TaFeng dataset, the analysis reveals that the model achieves its optimal performance with $L = 1$, indicating the significant influence of the most recent transaction. Although $L = 4$ shows a slightly higher HR@10, the overall metrics favor $L = 1$, suggesting that the latest transaction holds the most weight in influencing the next basket recommendation.

In contrast, the Android Apps dataset exhibits the best performance with a sequence length of $L = 5$. This result suggests that, within the tested range, longer sequence lengths contribute more significantly to the model's predictive accuracy. The increase in both HR and NDCG metrics as the sequence length extends to $L = 5$ supports the hypothesis that a broader context of user interactions leads to more accurate basket predictions.

TABLE 4.4: Sequence Length Analysis

Dataset	Metric	L=1	L=2	L=3	L=4	L=5
TaFeng	HR@10	0.3503	0.3487	0.3422	**0.3548**	0.3329
	HR@20	**0.4331**	0.4284	0.4207	0.4315	0.4096
	NDCG@10	**0.2538**	0.2474	0.2429	0.2519	0.2379
	NDCG@20	**0.2635**	0.2581	0.2550	0.2612	0.2480
Android Apps	HR@10	0.0505	0.0556	0.0684	0.0663	**0.0710**
	HR@20	0.0817	0.0897	0.1010	0.1059	**0.1100**
	NDCG@10	0.0308	0.0339	0.0395	0.0391	**0.0440**
	NDCG@20	0.0387	0.0425	0.0478	0.0492	**0.0540**

This analysis underscores the importance of considering the appropriate sequence length in next-basket recommendation models. While the immediate past transaction carries significant weight in certain contexts, a broader historical view can be more beneficial in others, as demonstrated by the distinct patterns observed across the two datasets.

4.8.4 Dimension Analysis

To understand the impact of latent dimension size on TransGAT's performance, a dimension analysis was performed using the TaFeng dataset. Due to constraints of GPU memory, the maximum dimension tested was 128 (64 for the T-Mall dataset). The analysis, encapsulated in Table 4.5, examines how different dimensional sizes, ranging from 8 to 128, affect the model's predictive capabilities.

The results indicate a clear trend: as the dimension size increases, the model's performance improves. This enhancement is evident across all evaluated metrics, including Hit Ratios (HR@10 and HR@20) and Normalized Discounted Cumulative Gains (NDCG@10 and NDCG@20). The improvement in performance with larger dimensions can be attributed to the enhanced representational capacity, allowing for more nuanced and detailed capturing of user-item interactions and preferences.

The analysis underscores the significance of choosing an appropriate dimension size for embedding layers in the context of next-basket recommendation. While larger dimensions can lead to increased computational requirements, they offer a

Table 4.5: Dimension Analysis

Dataset	Metric	dim=8	dim=16	dim=32	dim=64	dim=128
TaFeng	HR@10	0.3053	0.3222	0.3259	0.3306	**0.3503**
	HR@20	0.3815	0.3901	0.4046	0.4159	**0.4331**
	NDCG@10	0.2327	0.2404	0.2412	0.2405	**0.2538**
	NDCG@20	0.2436	0.2484	0.2525	0.2541	**0.2635**

substantial benefit in terms of model accuracy and effectiveness, as demonstrated by the consistent improvements observed with the increase in dimension size.

4.9 Summary

This chapter introduces **TransGAT**, a graph-neural-network-based inductive bias for next-basket recommendation, addressing the second research question from Section 1.2—*How to represent basket items' relationships in transitional sequences effectively for next basket recommendation?* TransGAT leverages *Graph Attention Networks (GATs)* to model transitions between items across consecutive baskets. The core *transitional graph attention layer* encodes item relationships by updating node embeddings based on neighbor importance through self-attention. A *transitional graph construction* module connects items appearing together across baskets, while basket representations are computed via average pooling of item embeddings. The model combines these with user embeddings in a prediction layer for next-basket recommendations. Extensive experiments demonstrate TransGAT's superior performance on real-world datasets. Ablation studies confirm the graph attention mechanism's importance for capturing basket transitions, while analyses of sequence length and embedding dimension highlight the value of properly modeling both recent and historical interactions. Overall, TransGAT provides an effective approach for modeling complex basket-level relationships in sequential recommendation.

Chapter 5

Multi-Scale Quasi-Transformer (MSQT)

5.1 Introduction

Recommending a sequence of items in one shot is essential for enhancing user engagement and aiding sellers' supply chain planning. In this context, the seq2seq[585] framework is employed, where user-item interactions serve as input to predict a sequence of items as output. While self-attention-based models have excelled in recommending single items, they often overlook group-level interactions among items, thus lacking in comprehensive representation. To address this, the ***M**ulti-**S**cale **Q**uasi-**T**ransformer* (**MSQT**) is introduced, an innovative model that extends the position-wise feed-forward network, akin to 1x1 convolution filters, in a multi-scale manner. This approach is designed to capture a variety of group-level features among self-attended items, thereby enhancing the model's understanding of complex item relationships. Furthermore, a novel temporal embedding method, termed ***D**ynamic **T**ime **W**arping **E**mbedding* (**DTWE**), is proposed to effectively incorporate temporal dynamics into the recommendation process. The DTWE method uniquely captures both the absolute timestamps of interactions and the relative temporal patterns within sequences, thus enriching the MSQT model's insight into user behavior over time and advancing the field of sequential recommendation.

FIGURE 5.1: Next-Items Recommendation Illustration

5.2 Problem Definitions and Formulations

The primary problem addressed in this work is **the recommendation of a sequence of items** (Figure 5.1) in one shot, a key challenge in the realm of sequential recommendation systems. This problem involves predicting a series of items that a user is likely to interact with next, based on their past interactions. This type of recommendation is crucial in various domains such as e-commerce, media streaming, and content browsing, where understanding and anticipating user preferences in a sequential manner is essential for enhancing user experience and engagement.

Given a user set U with $|U|$ users and an item set V with $|V|$ items, each user's interaction sequence S^u is presented as an ordered list of items $v \in V$. Each interaction sequence is denoted by $S^u = \{v_1^u, v_2^u, \ldots, v_L^u\}$ for user u, where L is the length of the sequence and v_i^u represents the i-th item interacted with by user u.

In this context, the next-items recommendation problem within a seq2seq framework is formulated as:

$$\text{Predict } \hat{S}^u_{\text{next}} = \{v^u_{L+1}, v^u_{L+2}, \ldots, v^u_{L+k}\} \text{ given } S^u = \{v^u_1, v^u_2, \ldots, v^u_L\} \text{ for } u \quad (5.1)$$

where $\hat{S}^u_{\text{next}}$ denotes the predicted sequence of the next k items for user u, and k is a predefined target sequence length, where $k \leq |V|$. The goal is to accurately predict the next k items in the sequence based on the user's past interactions, thereby enhancing the user experience and engagement in various applications.

Example in E-commerce: Consider an e-commerce platform where user u has previously interacted with the following items:

$$S^u = \{\text{laptop}, \text{mouse}, \text{keyboard}, \text{monitor}\}$$

The task is to predict the next sequence of items $\hat{S}^u_{\text{next}}$ that the user is likely to interact with. Based on the user's past interactions, the model might predict the next items as:

$$\hat{S}^u_{\text{next}} = \{\text{printer}, \text{USB drive}, \text{webcam}\}$$

Here, the predicted sequence helps in suggesting relevant items to the user, potentially increasing their engagement and satisfaction with the platform. Additionally, with the prediction of a sequence of items instead of a single item, the seller can better plan their supply chain and inventory management ahead, thus optimizing their operations.

Example in Media Streaming: On a media streaming platform, user u has watched the following sequence of videos:

$$S^u = \{\text{comedy movie}, \text{documentary}, \text{thriller movie}, \text{action movie}\}$$

The task is to predict the next sequence of videos $\hat{S}^u_{\text{next}}$ that the user is likely to watch. The model might predict the next videos as:

$$\hat{S}^u_{\text{next}} = \{\text{science fiction movie}, \text{romantic movie}, \text{animation}\}$$

This recommendation can improve the user's viewing experience by suggesting content that aligns with their interests based on their viewing history.

Example in Content Browsing: Consider a user u browsing through articles on a content website. The sequence of articles read by the user is:

$$S^u = \{\text{tech news}, \text{science article}, \text{health tips}, \text{travel blog}\}$$

The task is to predict the next sequence of articles $\hat{S}^u_{\text{next}}$ the user is likely to read. The model might predict the next articles as:

$$\hat{S}^u_{\text{next}} = \{\text{business news}, \text{lifestyle tips}, \text{sports update}\}$$

By providing such recommendations, the platform can keep the user engaged with relevant and interesting content, thereby enhancing their overall experience.

5.3 Contributions

The work presented in this chapter makes the following contributions, specifically tailored for the next-items recommendation task in a seq2seq setting, addressing the third research question outlined in Section 1.2: *How to effectively recommend a sequence of items in one shot through better modeling item sequences and time information?*

- **Multi-Scale Quasi-Transformer (MSQT):** A novel transformer-based encoder-decoder architecture for next-items recommendation in a seq2seq setting. MSQT integrates convolutional networks into the transformer structure, enabling multi-scale capture of group-level item interactions. This fusion facilitates the recognition of intricate patterns in user-item interaction sequences, thereby refining the recommendation process.
- **Dynamic Time Warping Embedding (DTWE):** An innovative temporal embedding method introduced to capture both absolute interaction timestamps and relative temporal patterns. DTWE enriches the sequential recommendation process by offering a deeper, more dynamic understanding of user behaviors and preferences over time.
- **Comprehensive Empirical Studies:** Extensive experiments conducted on four real-world datasets from Amazon Reviews. The MSQT model demonstrated superior performance, evidenced by higher Hit Ratios and Normalized Discounted Cumulative Gains (NDCGs) at various cutoffs, surpassing existing state-of-the-art methods. Additionally, the research included an ablation study and detailed analyses to explore the model's operational mechanics and validate the efficacy of the novel elements introduced.

Overall, the MSQT model's performance and the effectiveness of the DTWE method in capturing temporal dynamics demonstrate the potential of the proposed approach for recommending a sequence of items in various real-world settings.

5.4 Related Work

In this section, existing literature in three critical areas that form the foundation of this chapter are reviewed: next-items recommendation, the application of transformers in recommendation systems, and the integration of temporal information in sequential models.

5.4.1 Next-Items Recommendation

Next-items recommendation, a significant subfield of recommendation systems, focuses on predicting a sequence of future items based on past user interactions. This area has seen a shift from traditional collaborative filtering to advanced sequence modeling techniques, particularly utilizing sequence-to-sequence (seq2seq) setups. Earlier models often treated the recommendation as independent item predictions, overlooking the sequential dependencies. However, recent advancements emphasize capturing these sequential patterns, recognizing the importance of the order and context of past interactions. Techniques like Markov Chains and Recurrent Neural Networks (RNNs) have been instrumental in modeling user-item interactions more dynamically, significantly improving the ability to capture user preferences over time. The introduction of self-attention mechanisms has further revolutionized this field by effectively modeling dependencies and the importance of user short-term behavior patterns. Self-attention allows the model to weigh the relevance of each item in a user's interaction sequence, leading to more accurate predictions of future interactions. For example, [106] proposed a seq2seq training strategy based on latent self-supervision and disentanglement, which improves the recommendation accuracy by addressing the challenges of reconstructing future sequences. [170] emphasized the importance of modeling both past and future contexts to enhance session-based recommendations, using an encoder-decoder framework to fill gaps in user interaction sequences. [243] utilized adversarial learning in a seq2seq framework to better predict user behaviors, highlighting the robustness of this approach in handling sequential data. Similarly, [268] introduced a bidirectional encoder-decoder model that captures both forward and backward dependencies in user interaction sequences, further enhancing recommendation accuracy. [264] implemented recency sampling in seq2seq training to efficiently handle large-scale sequential data. [472, 533] also explored multiple contrastive signals

and multi-intention oriented contrastive learning within seq2seq setups to improve sequential recommendation performance.

5.4.2 Transformers in Sequential Recommendation

Transformers[30] have extended their impact from natural language processing to recommendation systems, primarily due to their self-attention mechanism. This feature allows transformers to weigh the importance of different items in a sequence, making them adept at handling sequential data. The architecture is particularly beneficial for recommendation systems, where understanding the sequence of user interactions is crucial. Innovations such as positional encodings and masked self-attention in transformers have shown substantial improvements in modeling complex user interaction patterns and longer user histories effectively. Bidirectional encoding approaches like BERT4Rec[69] enable the model to consider both past and future contexts, enhancing recommendation accuracy. Multi-temporal embeddings introduced in MEANTIME[127] allow the integration of various temporal features, thereby capturing the temporal dynamics of user behavior more precisely. Personalized transformers[123] tailor the model parameters to individual users, improving the personalization of recommendations. Techniques such as reverse pre-training[279] enhance the model's ability to understand the sequence of interactions by pre-training in the opposite order. Temporal graph models[202] and distribution modeling[203] incorporate temporal and probabilistic information, respectively, to better capture user interaction dynamics. Bridging NLP and recommendation systems through frameworks like Transformers4Rec[253] leverages the strengths of NLP techniques in understanding sequential data. Hypergraph enhancements[249] improve the modeling of complex relationships between items. Methods focusing on storage efficiency[220] reduce the computational load while maintaining performance. Integrating multi-behavior interactions[298, 327] and applying contrastive learning[211] help in capturing diverse user actions and distinguishing between similar user patterns. Denoising mechanisms[258] enhance the robustness of the model by mitigating noise in the data. Auxiliary relationships[319] and attention calibration[375] refine the attention mechanism to better focus on relevant items. Adaptive transformations[363, 401] adjust the model dynamically based on user interaction patterns. Hierarchical preference modeling[440] and offline reinforcement learning integration[427] offer advanced strategies for understanding user

preferences and learning from offline data. Preference editing[531] allows for the adjustment of recommendations based on user feedback. Sparse transformers[425] enhance efficiency by focusing on key interactions, and diversified recommendation approaches[545] ensure a broader range of recommendations, thus improving user satisfaction. These advancements collectively enhance the capability of transformers in capturing complex user interaction patterns and long-term dependencies, leading to more accurate and robust sequential recommendations.

5.4.3 Temporal Information Integration

Incorporating temporal dynamics is essential in sequential recommendation models to reflect the evolving nature of user preference. Traditional approaches often treated time as a static feature within models. However, more dynamic methods have been proposed for a better representation of temporal information. Techniques like Time-LSTM [83] introduce time gates, adding a layer of temporal sensitivity to the model by adjusting the input, forget, and output gates based on the time intervals between user interactions. While self-attention mechanisms have advanced sequential recommendations, models such as [16] do not explicitly address time representation but inherently consider the sequence of interactions. Other approaches, like those proposed by Ye [147], use a hierarchical recurrent neural network with a temporal attention mechanism that adjusts the influence of past interactions based on the time intervals, capturing both short-term and long-term dependencies. [127] leverages multi-temporal embeddings and a mixture of attention mechanisms to represent different time scales effectively. [20] introduces a time-interval-aware self-attention mechanism, adjusting attention scores based on the temporal proximity of interactions. [202] presents a continuous-time sequential recommendation model using a temporal graph collaborative transformer, capturing continuous-time dynamics with graph neural networks. [212] incorporates time lags between interactions, aiming to enhance the model's ability to capture temporal dynamics. [327] introduces a temporal graph transformer that models the temporal dependencies of multiple user behaviors. [410] discusses various methods for integrating temporal information into sequential recommendation models, emphasizing the importance of time-aware predictions. [288] presents RESETBERT4Rec, a pre-training model that combines temporal information with user historical behaviors using a modified BERT architecture. [534] proposes a position-enhanced and time-aware graph

convolutional network, integrating both sequential order and temporal intervals to improve recommendations. [140] introduces a knowledge and time-aware item modeling approach that captures temporal patterns alongside item knowledge. [345] offers a time-interval-aware data augmentation technique that uniformly samples time intervals to better represent the temporal distribution of interactions during training. Finally, [522] presents a method using graph ordinary differential equations (ODEs) to model continuous-time dynamics in sequential recommendations. These advancements underscore the critical role of temporal information in enhancing the accuracy and relevance of sequential recommendation models.

5.5 Methodology

This section describes the Multi-Scale Quasi-Transformer (MSQT) (Figure 5.2), a novel approach that integrates convolutional neural networks (CNNs) under multi-scale setup into the Transformer architecture for the task of next items recommendation. MSQT merges the multi-head self-attention mechanism of Transformers with the spatial feature extraction capabilities of CNNs. It is designed to capture both long-range dependencies and local features in sequence data, enhancing the model's ability to understand varying scales of item interactions and patterns in user behavior sequences. The model comprises an encoder-decoder architecture, with the encoder extracting features from the input sequence and the decoder predicting the next items in the sequence. The MSQT model consists of the following key components:

5.5.1 Item Representation

In the MSQT model, the representation of items is crucial for accurately capturing user interactions and their temporal dynamics. This representation is comprised of item embeddings, positional encodings, and Dynamic Time Warping Embeddings (DTWE), as depicted in Figure 5.3.

Each item in the sequence is initially represented by a one-hot encoded vector, which is then transformed into a dense vector through an embedding layer:

$$\mathbf{v}_i = \text{one-hot}(i), \quad \mathbf{x}_i = \mathbf{W}_{\text{emb}}\mathbf{v}_i \tag{5.2}$$

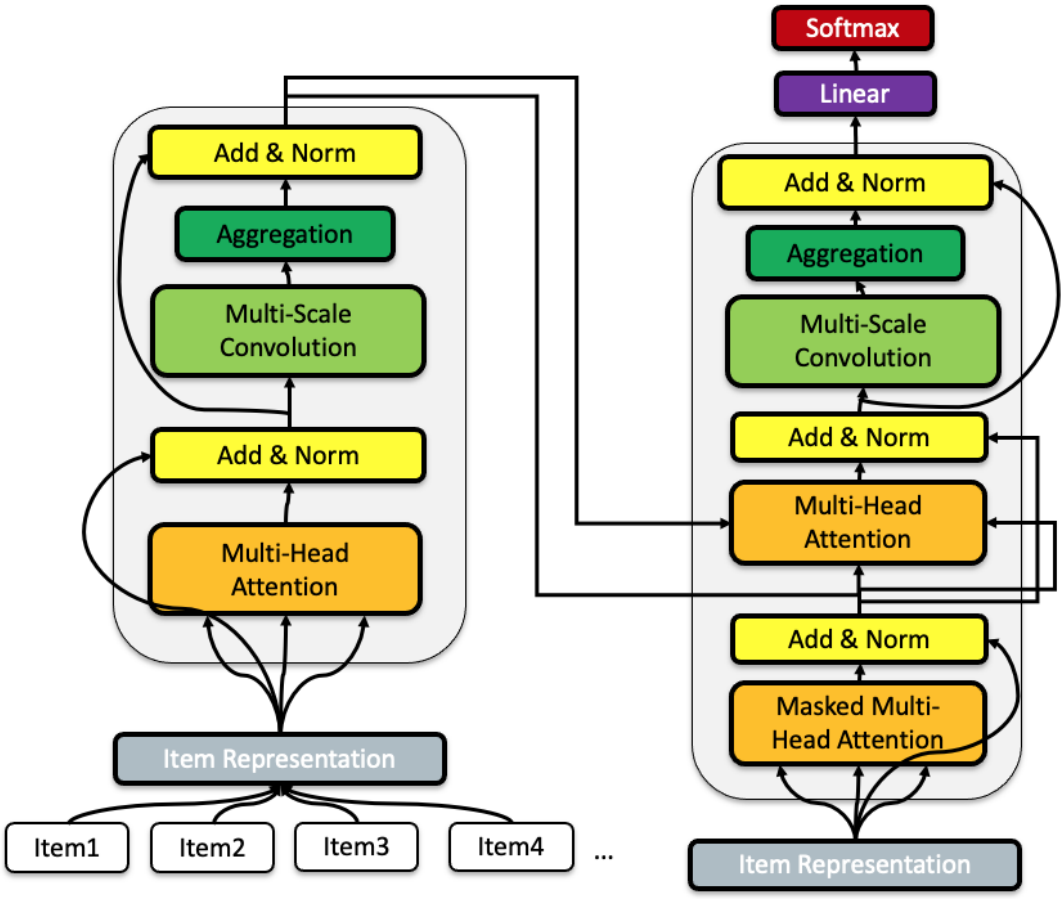


FIGURE 5.2: The Multi-Scale Quasi-Transformer (MSQT) Architecture

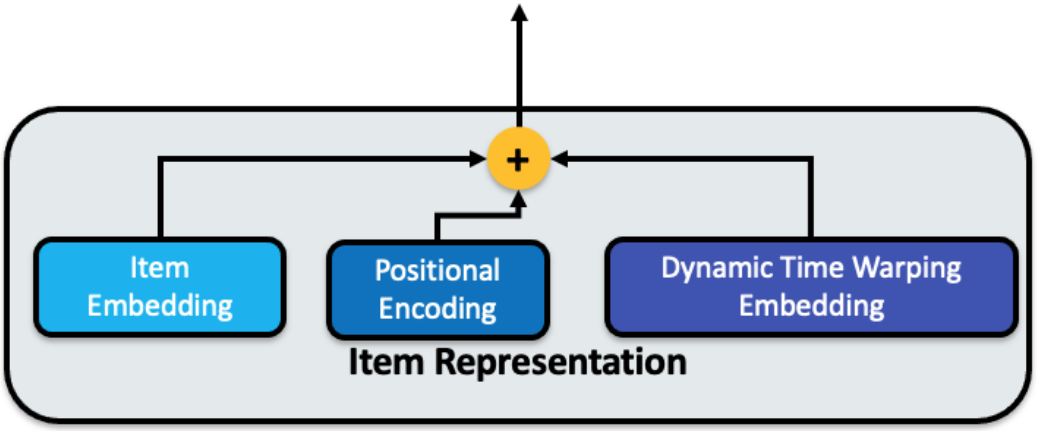


FIGURE 5.3: The Item Representation Module

where $\mathbf{v}_i$ denotes the one-hot vector of item i, with length equal to $|V|$, the number of items in the vocabulary. Here, $\mathbf{x}_i \in \mathbb{R}^{d_{\text{model}}}$ is the dense embedding vector of item i, and $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{d_{\text{model}} \times |V|}$ is the embedding weight matrix. The variable d_{model} represents the embedding dimension, and $|V|$ denotes the number of items in the vocabulary.

Positional encoding is added to account for the order of items within the sequence:

$$\mathbf{PE}_{(\text{pos},2i)} = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right), \quad \mathbf{PE}_{(\text{pos},2i+1)} = \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \tag{5.3}$$

where pos indicates the position in the sequence, i is the dimension index, and d_{model} is the dimension of the embedding.

Dynamic Time Warping Embedding (DTWE) significantly enhances the representation of temporal dynamics in sequential recommendation systems by using a warping path that aligns time points across item sequences. This method is particularly beneficial for capturing the temporal patterns and variations in user-item interactions, which occur at different times and speeds, thus providing a more accurate reflection of changing user preferences:

$$\mathbf{DTWE} = \sum_{(i_k, j_k) \in \mathbf{P}} \exp\left(-\frac{||t_{i_k} - t_{j_k}||}{\boldsymbol{\sigma}}\right) \tag{5.4}$$

In this approach, $\mathbf{P}$ represents the optimal warping path, consisting of pairs (i_k, j_k), which aligns timestamps t_{i_k} from one item sequence with t_{j_k} from another based on the minimum distance criterion. This criterion reflects the least temporal discrepancy between aligned items, enabling the model to adaptively respond to diverse timing of events within user sequences.

For instance, consider two users' interaction sequences: one views products A, B, C in January and products X, Y, Z in February, while another views A, Y, B, Z within a single month. DTWE can align A with A, and B with B, while C from the first sequence might align with Y from the second, suggesting a potential interest linkage despite the temporal gap. Similarly, Z aligns directly as it appears in both users' sequences, even if at different times.

The parameter $\boldsymbol{\sigma}$, a vector of sensitivities for each temporal dimension, dictates how temporal discrepancies influence the embeddings of items. A smaller $\boldsymbol{\sigma}$ makes

the embedding sensitive to smaller time differences, suitable for capturing short-term behavior patterns, whereas a larger $\boldsymbol{\sigma}$ accommodates longer-term trends, providing a comprehensive view of user preferences over extended periods. This adaptive alignment helps improve the accuracy and relevance of recommendations, effectively managing the complex dynamics of user interactions.

The comprehensive item representation in the MSQT model combines these components into a unified embedding:

$$\mathbf{E}_{\text{item}} = \mathbf{X} + \mathbf{PE} + \mathbf{DTWE} \tag{5.5}$$

where $\mathbf{X}$ is the item embedding matrix, $\mathbf{PE}$ is the positional encoding matrix, and $\mathbf{DTWE}$ is the Dynamic Time Warping Embedding matrix. This representation encapsulates the item's features, position in the sequence, and temporal dynamics, providing a comprehensive input for the subsequent layers of the model.

5.5.2 Multi-Scale Quasi-Transformer (MSQT): Encoder

The encoder in MSQT comprises a multi-head self-attention module, a multi-scale convolutional module, and a. The encoder's role is to extract features from the input sequence, providing a representation of the sequence that is conducive to the decoder's predictive process.

Multi-Head Self-Attention Module. The MSQT model incorporates a multi-head self-attention module, an enhancement of the standard Transformer self-attention, to process sequences. This module allows the model to concurrently process information from different representation subspaces at various positions. It is instrumental in capturing complex dependencies and diverse aspects of the input sequence.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{5.6}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{5.7}$$

In these equations, head_i represents the i^{th} attention head, and W_i^Q, W_i^K, and W_i^V are the respective weight matrices for the query, key, and value of the i^{th} head. W^O is the output weight matrix that combines the outputs of all heads. This

modular approach enables the model to extract and utilize diverse features from the sequence data effectively.

Add and Norm. The output of the self-attention module is added to its input and then normalized:

$$\mathbf{H}_{\text{norm}} = \text{LayerNorm}(\mathbf{H} + \text{MultiHead}(\mathbf{H})) \tag{5.8}$$

where $\mathbf{H}$ is the input to the self-attention module.

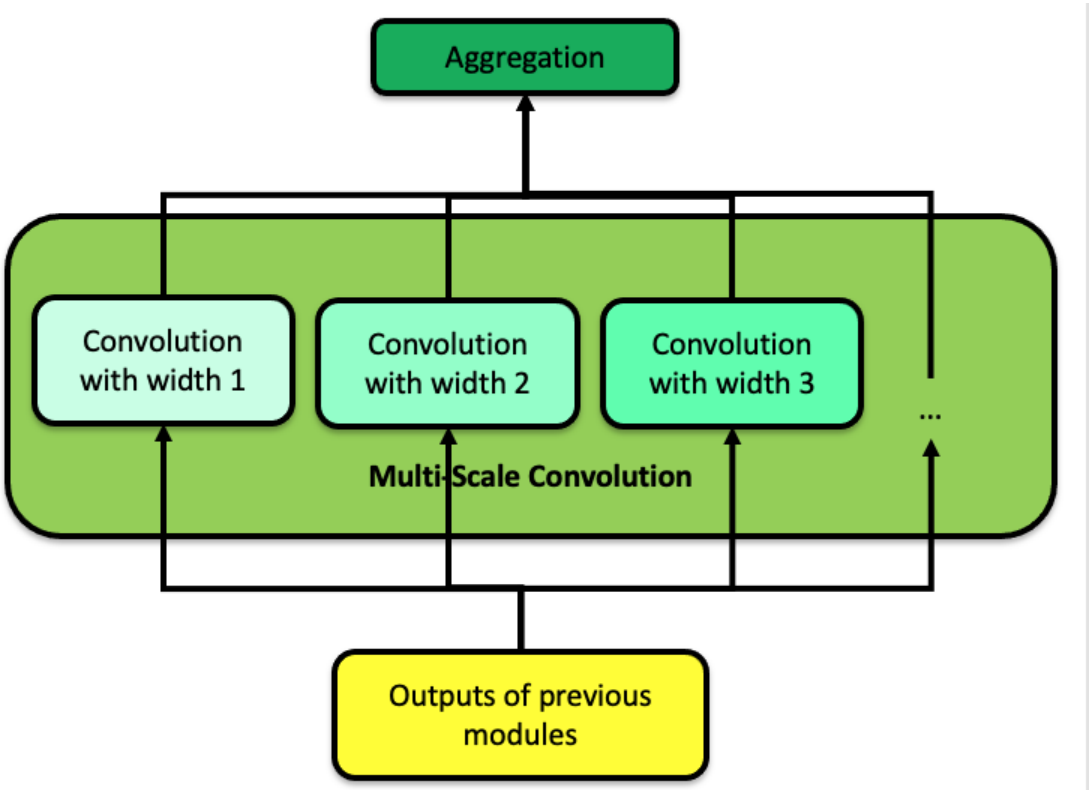


FIGURE 5.4: The Multi-Scale Convolution Module

Multi-Scale Convolutional Module. Following the multi-head self-attention module (Figure 5.4), the MSQT model employs a multi-scale convolutional module to further enhance feature extraction. This module comprises convolutional layers with various kernel sizes, allowing the model to detect patterns and features at multiple scales. This capability is crucial for identifying both immediate and extended dependencies in sequential data.

$$\mathbf{C}_l = \text{ReLU}(\mathbf{W}_l * \mathbf{H} + \mathbf{b}_l) \tag{5.9}$$

where $\mathbf{C}_l$ is the output of the l^{th} convolutional layer, $\mathbf{W}_l$ and $\mathbf{b}_l$ represent the layer's weights and biases, respectively, and $\mathbf{H}$ is the output from the multi-head self-attention module. The integration of the multi-scale convolutional module significantly boosts the MSQT's ability to discern complex patterns in the data, complementing the global contextual insights provided by the self-attention module.

Aggregation. The outputs from the convolutional module are aggregated to form a comprehensive vector representation of the sequence. This aggregation is crucial for synthesizing the multi-scale features extracted by the convolutional layers, thereby enhancing the model's overall understanding of the sequence:

$$\mathbf{C}_{\text{agg}} = \text{Concat}(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_l) \tag{5.10}$$

Here, $\mathbf{C}_l$ represents the output of the l^{th} convolutional layer. The Concat function combines these outputs into a single vector, $\mathbf{C}_{\text{agg}}$, which encapsulates the comprehensive feature set extracted from the sequence. This aggregated representation is instrumental in facilitating the subsequent layers of the model, particularly the decoder's predictive process.

Add and Norm. Similar to the self-attention module, the output of the convolutional module is added to its input and normalized:

$$\mathbf{C}_{\text{final}} = \text{LayerNorm}(\mathbf{C}_l + \mathbf{H}_{\text{norm}}) \tag{5.11}$$

where $\mathbf{C}_l$ is the output of the convolutional module, and $\mathbf{H}_{\text{norm}}$ is the normalized output from the self-attention module.

5.5.3 Multi-Scale Quasi-Transformer (MSQT): Decoder

The decoder in the MSQT model is designed to generate a sequence of recommended items based on the user's historical interactions. It employs a similar architecture to the encoder, featuring a multi-head self-attention module, a multi-scale convolutional module, and a temporal embedding layer. The decoder's primary function is to predict the next item in the sequence by attending to the relevant information from the user's history and the items' contextual representations.

Multi-Head Self-Attention Module. The decoder's multi-head self-attention module allows it to capture dependencies and relationships between the items in the output sequence. It enables the decoder to consider the context of the previously generated items when predicting the next item in the sequence.

$$\text{MultiHead}(Q', K', V') = \text{Concat}(\text{head}'_1, \text{head}'_2, \ldots, \text{head}'_h)W^{O'} \tag{5.12}$$

$$\text{where head}'_i = \text{Attention}(Q'W_i^{Q'}, K'W_i^{K'}, V'W_i^{V'}) \tag{5.13}$$

Here, Q', K', and V' are the query, key, and value matrices for the decoder, with respective weight matrices $W_i^{Q'}$, $W_i^{K'}$, and $W_i^{V'}$ for each head. $W^{O'}$ is the output weight matrix.

Multi-Scale Convolutional Module. The decoder's multi-scale convolutional module is employed to capture local item patterns and relationships at different scales within the generated sequence. This module enhances the decoder's ability to consider the context of the previously generated items when predicting the next item.

$$\mathbf{C}'_l = \text{ReLU}(\mathbf{W}'_l * \mathbf{H}' + \mathbf{b}'_l) \tag{5.14}$$

In this equation, $\mathbf{C}'_l$ denotes the output of the l^{th} convolutional layer in the decoder, with $\mathbf{W}'_l$ and $\mathbf{b}'_l$ as the layer's weights and biases, and $\mathbf{H}'$ as the input from the decoder's multi-head self-attention module.

Attention Mechanism. The decoder employs an attention mechanism to focus on the relevant items from the user's historical interactions when generating the output sequence. The attention mechanism computes a weighted sum of the encoder's output, where the weights are determined by the compatibility between the decoder's hidden state and the encoder's output at each position.

$$\mathbf{A}' = \text{softmax}(\mathbf{H}'\mathbf{W}'_a\mathbf{H}^T) \tag{5.15}$$

$$\mathbf{C}' = \mathbf{A}'\mathbf{H} \tag{5.16}$$

Here, $\mathbf{H}'$ is the decoder's hidden state, $\mathbf{H}$ is the encoder's output, $\mathbf{W}'_a$ is a learnable weight matrix, and $\mathbf{C}'$ is the context vector that captures the relevant information from the user's historical interactions.

Output Generation. The decoder generates the output sequence one item at a time. At each step, the decoder takes the previous item's embedding, the context vector from the attention mechanism, and the decoder's hidden state as input and produces a probability distribution over the item vocabulary.

$$\mathbf{y}'_t = \text{softmax}(\mathbf{W}'_o[\mathbf{x}'_{t-1}; \mathbf{c}'_t; \mathbf{h}'_t] + \mathbf{b}'_o) \tag{5.17}$$

where $\mathbf{y}'_t$ is the output probability distribution at time step t, $\mathbf{x}'_{t-1}$ is the embedding of the previous item, $\mathbf{c}'_t$ is the context vector, $\mathbf{h}'_t$ is the decoder's hidden state, and $\mathbf{W}'_o$ and $\mathbf{b}'_o$ are the output layer's weights and biases.

5.5.4 Model Training

The MSQT model is trained based on the binary cross-entropy loss. The loss function is defined as:

$$l = \sum_u (\sum_i -\log(\sigma(y^u_{i,t})) + \sum_j -log(1 - \sigma(y^u_{j,t}))) \tag{5.18}$$

where σ denotes the sigmoid function, and i and j represent target and negative items, respectively. The model is trained to minimize this loss function, optimizing the parameters to improve the prediction accuracy of the next items in the sequence. A 3:1 negative sampling ratio is adopted, following [1, 2, 11].

Optimization. The model is optimized using the Adam optimizer, which adapts the learning rate for each parameter based on its historical gradients. The Adam optimizer helps the model converge faster and reduces the impact of the learning rate choice.

Regularization. To prevent overfitting, the model employs L2 regularization on the model's weights. L2 regularization adds a penalty term to the loss function, discouraging the model from learning large weights and promoting more generalizable solutions.

$$L_{\text{reg}} = \lambda \sum_i \|\mathbf{W}_i\|_2^2 \tag{5.19}$$

where λ is the regularization coefficient, and $\mathbf{W}_i$ are the model's weight matrices.

5.6 Experiment Setup

This section outlines the datasets, baselines, and evaluation protocols used in the empirical studies conducted to evaluate the MSQT model's performance.

5.6.1 Datasets

Four datasets from the **Amazon Reviews** collection[1] are utilized, chosen for their characteristic sparsity and diversity. Amazon, a globally recognized e-commerce platform, provides a rich repository of user interactions through product reviews and ratings. This study focuses on user-item interaction patterns, utilizing ratings as indicators of user preferences, rather than analyzing review content [651–653]. Similar as for Chapter 3, due to time constraint, 4 such datasets are selected and they are highly suitable for next items recommendation due to their extensive user interactions and diverse product categories, offering a robust foundation for evaluating recommendation models. To ensure the relevance and quality of the datasets, the following preprocessing steps were implemented to refine the datasets for this analysis:

- **Interactions with user ratings of 3 or above are considered:** This establishes a baseline for user preference, ensuring that the interactions taken into account reflect positive user experiences, which are more indicative of genuine user interests.
- **Chronological order is determined using time-related attributes:** This is essential for next item recommendation tasks as it aids in comprehending the sequence of interactions, allowing the model to predict future interactions based on past behavior.
- **Users with less than 10 interactions are excluded:** This criterion focuses on longer user sequences, providing richer data for training and addressing the cold start problem, a common issue in recommendation systems [1, 11, 651, 652, 654, 655]. This ensures that the model can learn more meaningful patterns from substantial user histories.

[1] http://jmcauley.ucsd.edu/data/amazon/

The preprocessing protocols employed are in line with standard practices in the field and aim to enhance the validity and reliability of the model's performance in real-world contexts, as well as better for the next items recommendation task. The datasets used in this study are summarized in Table 5.1.

TABLE 5.1: Dataset Statistics

Dataset	**Users**	**Items**	**Interactions**	**Sparsity (%)**
Electronics	1,530	29,381	101,436	99.77
Tools/Home	2,908	9,565	45,786	99.84
Cell Phones	2,440	8,152	36,246	99.82
Beauty	4,204	11,164	75,103	99.84

5.6.2 Baselines

The proposed model is evaluated against 9 baselines available at the time this research was conducted to ensure a comprehensive comparison. The baselines span traditional collaborative filtering, matrix factorization, neural networks, and advanced deep learning approaches. Except for PopRec, PersonalPopRec, and kNN, all the other established baselines are implemented in a seq2seq manner. The following baselines are included in the evaluation:

PopRec: This method recommends items based on their overall popularity across the dataset, gauging frequency as a measure of general user interest. Under seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in the sequence by leveraging the popularity of items.

PersonalPopRec: Tailoring recommendations to individual users, this approach ranks items according to their frequency in a particular user's interaction history. In a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in the sequence by considering the user's personalized preferences.

kNN: A classic collaborative filtering method, k-Nearest Neighbors, relies on similarity metrics between users to generate recommendations. In a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in the sequence by identifying similar users and leveraging their interactions.

FPMC [1]: This model merges Matrix Factorization (MF) with First-order Markov Chains (FMC), adapting it for next-basket recommendation scenarios. For our purposes, each basket is treated as containing a single item. To make it into a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in a sequence given the previous items by incorporating sequential dependencies.

MLP [6]: Employing a traditional Multi-Layer Perceptron architecture, this method is a neural network-based approach for next-item recommendations. To make it into a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in a sequence given the previous items by leveraging the network's ability to capture non-linear relationships.

GRU4Rec [6]: This Recurrent Neural Network (RNN)-based model is designed for session-based recommendations, utilizing Gated Recurrent Units (GRUs). To adapt it into a seq2seq setup, the GRU4Rec model predicts the next item using the last predicted item with previous items continuously in the sequence by capturing the temporal dependencies between user interactions.

Caser [11]: This Convolutional Neural Network (CNN)-based model targets next-item recommendation, applying convolutional operations to capture sequential patterns. For a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously by leveraging the sequential patterns identified through convolutional filters.

SASRec [16]: A model grounded in self-attention mechanisms, SASRec is tailored for next-item recommendations, offering an advanced approach to sequence modeling. In a seq2seq setup, SASRec predicts the next item using the last predicted item with previous items continuously by focusing on the most relevant previous items using self-attention.

Transformer [30]: Renowned for its self-attention and point-wise feed-forward network, the Transformer model is a highly influential architecture in sequence-based recommendation tasks. To convert it into a seq2seq setup, the Transformer predicts the next item using the last predicted item with previous items continuously by capturing complex dependencies across the entire sequence through its self-attention mechanism.

These baselines represent a diverse spectrum of methodologies in the field of recommendation systems, ranging from classical algorithms to state-of-the-art deep learning models. They provide a robust context for evaluating the performance and contributions of the proposed approach.

5.6.3 Evaluation Protocols

Evaluating the effectiveness of the model is crucial for understanding its performance and potential impact in real-world recommendation scenarios. The evaluation protocols employed in this study are designed to provide a comprehensive assessment of the model's efficacy in predicting the next items in user sequences. Since the setup is a seq2seq model, the final evaluation results are averaged over the target length.

Metrics: Two pivotal metrics, Hit Ratio (HR) and Normalized Discount Cumulative Gain (NDCG), are employed to evaluate the model's ranking efficacy at cut-off points 10 and 20. In a seq2seq setup, the model predicts the next item using the last predicted item with previous items continuously in the sequence. The evaluation compares the predicted item with the actual next item in the sequence, averaged over the predicted sequence length.

- **Hit Ratio (HR):** Measures if the actual item is within the top-K recommendations. A higher HR@K indicates better model precision in capturing user preferences.
- **Normalized Discount Cumulative Gain (NDCG):** Evaluates the ranking quality by considering the position of the correct item within the recommended list. Higher NDCG@K scores indicate more effective ranking of relevant items.

These metrics ensure a comprehensive evaluation of the model's performance, providing a nuanced understanding of its effectiveness in real-world recommendation scenarios.

5.6.4 Implementations

The Multi-Scale Quasi-Transformer (MSQT) and baseline models were implemented using Pytorch [656], incorporating aspects from original authors' versions when needed. Experiments were conducted on Nvidia V100 and RTX 2080ti GPUs. Hyperparameters were first optimized via grid search, then kept constant for subsequent runs. Testing set results, corresponding to the best validation metrics, were averaged over the 10 runs and reported. Early stopping was applied during training at the point of validation loss and metric plateau.

In terms of hyperparameter tuning, the optimizer for all models was Adam [631], selected for its efficiency. The learning rate was tuned within {1e-3, 3e-4, 1e-4} and finally set at 1e-3. L2 regularization was adjusted within {1e-1, 1e-2, ..., 1e-6} and fixed at 1e-6. The latent dimension, crucial for the model's performance, was tested from {16, 32, 64, 128, 256} and finalized at 128. Batch size, a key factor in model training, was set at 256 from choices {16, 32, 64, 128, 256}. Dropout [657], added to enhance model generalization, was tuned between {0.2, 0.5}. Model parameters were initialized using Xavier initialization [659] or according to original model settings, depending on the requirement.

For MSQT, the number of blocks was a crucial parameter and was set after testing within {1, 2, 3, 4} and the scale from {1, 2, 3, 4, 5}. MLP's number of layers was similarly adjusted from {1, 2, 3, 4}. For Caser, horizontal filters were selected from {4, 8, 16, 32, 64}, vertical filters from {2, 4, 6, 8}, and union-level length within {1, 2, ..., 9}. GRU4Rec's layer count was crucial for its performance and was tuned from {1, 2, 3, 4}. For SASRec, the number of blocks, affecting model depth, was tested within {1, 2}. The Transformer model's number of blocks, key to its structure, was explored within {1, 2, 3, 4}. All these hyperparameters were carefully selected via grid search based on validation set performance and maintained constant for consistency in the remaining experimental runs.

5.7 Experiment Results and Analysis

This section is devoted to evaluating the effectiveness of MSQT, the proposed model for the next items recommendation task. The analysis to address the following research questions are structured as below:

RQ1: Does MSQT outperform state-of-the-art methods in next-items recommendation across various datasets?

RQ2: How does MSQT's performance dependent on various components? How is ablation study?

RQ3: How does scale of the convolutional kernel affect MSQT's performance?

RQ4: How is the sigma in the temporal embedding layer affect MSQT's performance?

5.7.1 Recommendation Performance

The performance of the Multi-Scale Quasi-Transformer (MSQT) in comparison to various baseline models highlights its effectiveness in the domain of next-items recommendation. The relative performance improvement of MSQT over the second-best method is quantified as follows:

$$\textbf{Improvement} = \textit{MSQT} - \textit{second best performer} \tag{5.20}$$

$$\textbf{Rel Gain} = \frac{\textbf{Improvement}}{\textit{second best performer}} \times 100\% \tag{5.21}$$

TABLE 5.2: Performance of Models Across Selected Datasets

Dataset	Metric	PR	PPR	kNN	FPMC	MLP	GRU4R	Caser	SASRec	Transf	MSQT	Rel Gain
Electronics	HR@10	0.1818	0.2087	0.2199	0.2160	0.2306	0.2197	0.2306	0.1686	0.2037	**0.2406**	4.31%
	NDCG@10	0.1151	0.1763	0.1751	0.1594	0.1180	0.1678	0.1892	0.1692	0.1892	**0.1992**	5.29%
	HR@20	0.2880	0.2328	0.2716	0.2706	0.3285	0.2935	0.3599	0.3622	0.3555	**0.3722**	3.42%
	NDCG@20	0.2192	0.1505	0.1897	0.1929	0.2419	0.2366	0.2458	0.1990	0.1717	**0.2519**	2.49%
Tools/Home	HR@10	0.2067	0.2178	0.2381	0.1635	0.2476	0.2447	0.1665	0.1727	0.2163	**0.2576**	4.04%
	NDCG@10	0.1519	0.1688	0.1805	0.1556	0.1220	0.1109	0.1562	0.1146	0.1860	**0.1905**	2.42%
	HR@20	0.2672	0.2678	0.3114	0.2718	0.3698	0.3498	0.3170	0.2656	0.3700	**0.3798**	2.65%
	NDCG@20	0.1594	0.2462	0.2193	0.2091	0.2368	0.1860	0.1738	0.1934	0.2365	**0.2562**	8.34%
Cell Phones	HR@10	0.2076	0.1749	0.2225	0.2074	0.1662	0.2230	0.2434	0.1812	0.1820	**0.2534**	4.11%
	NDCG@10	0.1355	0.1574	0.1865	0.1622	0.1199	0.1517	0.1529	0.1402	0.1924	**0.1965**	2.13%
	HR@20	0.3601	0.3107	0.3002	0.3214	0.3162	0.2540	0.3160	0.3275	0.3522	**0.3701**	5.08%
	NDCG@20	0.2167	0.2072	0.2144	0.1931	0.2348	0.1554	0.2090	0.1680	0.2331	**0.2448**	5.01%
Beauty	HR@10	0.1632	0.1723	0.1924	0.2397	0.2434	0.1700	0.2230	0.1525	0.2129	**0.2497**	2.59%
	NDCG@10	0.1332	0.1458	0.1115	0.1449	0.1426	0.1735	0.1419	0.1644	0.1835	**0.1965**	7.11%
	HR@20	0.3303	0.3634	0.3149	0.2815	0.3097	0.2326	0.2737	0.2394	0.3522	**0.3734**	6.02%
	NDCG@20	0.1683	0.2346	0.1802	0.2392	0.1569	0.1724	0.1710	0.1954	0.2425	**0.2492**	2.76%

As demonstrated in Table 5.2, MSQT consistently surpasses other models across all datasets in terms of HR and NDCG at both 10 and 20 cutoffs. Notably, in the Electronics dataset, MSQT shows a 4.31% improvement in HR@10 over its closest competitor, signifying its capability to accurately capture user preferences.

Additionally, in the NDCG@20 metric, MSQT achieves a 2.49% improvement, indicating its effectiveness in ranking relevant items higher in the recommendation list. In the Tools/Home dataset, MSQT demonstrates its strength with a 4.04% increase in HR@10 and a significant 8.34% improvement in NDCG@20. These results highlight MSQT's ability to understand and respond to the diverse preferences of users in this category. Similarly, in the Cell Phones dataset, MSQT's improvements in HR@20 and NDCG@20, at 5.08% and 5.01% respectively, underscore its proficiency in providing relevant recommendations in this rapidly changing and varied market. In the Beauty dataset, MSQT outperforms other models with a 2.59% increase in HR@10 and a notable 7.11% increase in NDCG@10. This suggests that MSQT is particularly adept at handling the nuanced preferences typical in beauty product selections. Overall, these results confirm the effectiveness of MSQT in the realm of sequential recommendation. Its ability to integrate multi-scale convolutional features and dynamic temporal embeddings allows it to outperform traditional models, especially in terms of improving the ranking of recommended items, thus providing more accurate and relevant recommendations across various consumer categories.

5.7.2 Component Analysis and Ablation Study

In this subsection, how different components of MSQT contribute to its overall performance is investigated. An ablation study is conducted to understand the impact of each component. The study involves systematically removing one component at a time and observing the change in performance. The results are summarized in Table 5.3.

Table 5.3: Ablation Study Results on Electronics Dataset

Component	HR@10	NDCG@10	HR@10 Delta %	NDCG@10 Delta %
Full Model	0.2534	0.1965	-	-
w/o Convolutional Layers	0.2373	0.1787	-6.35	-9.03
w/o DTWE	0.2428	0.1873	-4.17	-4.71

The ablation study of the MSQT model reveals the significant contribution of its individual components to its overall performance. Removing the convolutional layers results in a decline of 6.35% in HR@10 and a more pronounced 9.03% drop

in NDCG@10, underscoring the layers' role in capturing complex patterns in the data. The removal of DTWE, on the other hand, leads to a 4.17% decrease in HR@10 and a 4.71% reduction in NDCG@10, indicating its importance in effectively integrating temporal dynamics. These results highlight the convolutional layers' critical function in enhancing the model's precision, while DTWE plays a key role in ensuring the model's relevance and timeliness in recommendations. The varied impacts of these components underscore their synergistic effect in MSQT, with each element contributing uniquely to the model's robustness and accuracy in predicting user preferences.

5.7.3 Effect of MSQT Scale

This subsection evaluates the impact of different MSQT scale configurations on the performance of the model in the context of an electronics dataset. Experiments are conducted using various scale settings of the MSQT model, and the results of these experiments are presented in Table 5.4.

TABLE 5.4: Impact of MSQT Scale Setups on Electronics Dataset

Scale	HR@10	NDCG@10
MSQT(w=1)	0.2623	0.2011
MSQT(w=2)	<u>0.2678</u>	<u>0.2056</u>
MSQT(w=3)	0.2564	0.1983
MSQT(w=4)	0.2597	0.2005
MSQT	**0.2754**	**0.2120**
Rel Gain (%)	2.85	3.11

The investigation into the MSQT model's scale variations reveals significant insights into their effect on the recommendation system's accuracy. The model's performance metrics, HR@10 and NDCG@10, indicate varying levels of effectiveness across different scale settings. The configuration MSQT(w=1) shows a solid baseline performance. However, it is the MSQT(w=2) setup that emerges as the second-best performer, slightly elevating the HR@10 and NDCG@10 metrics compared to the baseline. The configurations MSQT(w=3) and MSQT(w=4) exhibit a minor decline in performance, suggesting that these settings might not capture the dataset's nuances as effectively as MSQT(w=2).

Most notably, the integrated MSQT configuration outperforms all individual scale settings, achieving the highest HR@10 and NDCG@10 scores. The relative gain

in performance of the overall MSQT configuration over the second-best performer (MSQT(w=2)) is quantified as 2.85% and 3.11% in HR@10 and NDCG@10, respectively. This indicates that the comprehensive MSQT configuration is more adept at capturing the intricate patterns necessary for precise user preference prediction. These findings highlight the importance of scale configuration in sequential recommendation models, emphasizing the balance between model complexity and predictive accuracy in recommendation systems.

5.7.4 Impact of Sigma in DTWE

Here, how the sigma parameter in the DTWE temporal embedding layer influences MSQT's performance is explored. Various sigma values are tested, and the outcomes are presented in Table 5.5.

Table 5.5: Impact of Sigma in DTWE on Electronics Dataset

Sigma Value	HR@10	NDCG@10	HR@10 Delta %	NDCG@10 Delta %
0.5	0.2534	0.1965	-	-
1.0	0.2488	0.1856	-1.82	-5.55
2.0	0.2380	0.1940	-6.08	-1.27

The exploration of different sigma values in the DTWE temporal embedding layer of the MSQT model highlights their significant influence on the model's performance. The baseline sigma value of 0.5, which yields HR@10 and NDCG@10 scores of 0.2534 and 0.1965 respectively, serves as the optimal setting. Increasing the sigma to 1.0 results in a modest reduction in HR@10 by 1.82% and a more notable decline in NDCG@10 by 5.55%, indicating a sensitivity of the model's performance to this parameter. A further increase in sigma to 2.0 exacerbates the performance drop, with HR@10 decreasing by 6.08% and NDCG@10 showing a smaller decrease of 1.27%. These trends suggest that while larger sigma values may potentially broaden the temporal scope, they can also dilute the precision and relevance of the temporal embeddings, thereby impacting the model's overall efficacy. The findings underscore the importance of carefully tuning the sigma parameter in DTWE to balance between capturing wider temporal patterns and maintaining the accuracy and relevance of recommendations.

5.8 Summary

This chapter introduced the **Multi-Scale Quasi-Transformer (MSQT)** model, a novel neural inductive bias for next-items recommendation in a seq2seq setting, addressing the third research question: *How to effectively recommend a sequence of items in one shot through better modeling item sequences and time information?* The MSQT model integrates convolutional networks into a transformer-based encoder-decoder architecture, enabling the capture of multi-scale item interactions—an essential factor for recognizing intricate patterns in user-item sequences. Another key contribution of MSQT is the *Dynamic Time Warping Embedding* (DTWE), a temporal embedding method that captures both absolute interaction timestamps and relative temporal patterns. This design choice enriches the sequential recommendation process by offering a deeper understanding of user behaviors and preferences over time. The multi-scale convolutional module, combined with DTWE, empowers the MSQT architecture to effectively handle diverse time intervals between interactions, crucial for modeling user preferences that evolve at different speeds. By merging self-attention with multi-scale convolutional layers, MSQT balances global contextual awareness with localized pattern extraction. Experimental results on real-world Amazon Reviews datasets demonstrate its superior performance across key ranking metrics such as Hit Ratio and NDCG, thus underscoring the effectiveness of its design.

Chapter 6

Temporal Sets Nested Transformer (TSNT)

6.1 Introduction

The field of recommendation systems is rapidly evolving, with a growing focus on the subtask of temporal sets recommendation. This area aims to predict future user engagements based on historical interactions with sequences of item sets, each distinguished by unique timestamps. To address this complex challenge, the ***T**emporal **S**ets **N**ested **T**ransformer* (**TSNT**) is introduced, representing a significant stride in processing sequences of item sets within a temporal context. **TSNT** adopts an innovative approach to process item sets, moving away from the conventional practice of directly embedding items within a transformer framework. Instead, the process starts with an Set Items Transformer (**SIT**) for encoding each set. This crucial step ensures the generation of rich, detailed set embeddings that effectively capture the intricacies within each item set. These set embeddings are then seamlessly integrated into a more advanced transformer, the Temporal Sets Transformer (**TST**). **TST** is specially designed for the effective assimilation and interpretation of set sequence data with timestamps. A notable feature of **TSNT** is the introduction of Time Interval Quantizational Encoding (**TIQE**), an inventive relative position encoding method that complements the conventional Positional Encoding. **TIQE** is strategically designed to infuse the temporal context into the transformer architecture by embedding the quantized time intervals among sets,

significantly improving the model's proficiency in discerning temporal relationships between sets.

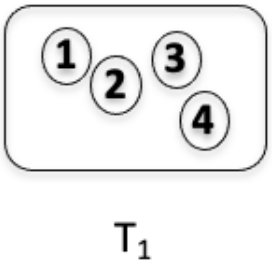


FIGURE 6.1: A Temporal Set

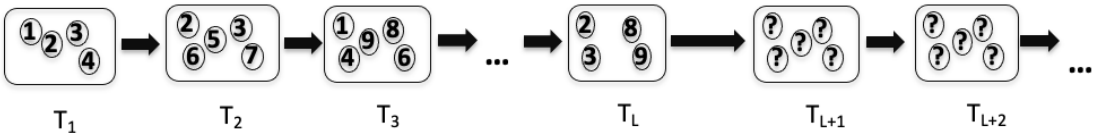


FIGURE 6.2: Temporal Sets Recommendation Illustration

6.2 Problem Definitions and Formulations

This section introduces key concepts and formalizes the Temporal Sets Recommendation problem. This problem is derived from the conventional temporal sets prediction problem[28], with a focus on capturing the dynamic nature of user interactions with items over time, represented as temporal sets. The definitions and formulations presented here lay the groundwork for understanding the challenges and objectives addressed by the Temporal Sets Nested Transformer (**TSNT**).

Definition 1: Temporal Set

A *Temporal Set* (Figure 6.1) is defined as a collection of unordered items that are associated with the same timestamp. Such sets can manifest in various forms, including shopping baskets, items engaged with during a single website visit, or acquisitions over defined timeframes like daily purchases.

Definition 2: Temporal Sets Recommendation

Temporal Sets Recommendation (Figure 6.2) refers to the process of forecasting future sets of items for users, based on their past interactions with similar sets. This prediction considers not only the items themselves but also the timestamps when these interactions occurred.

Formulation of the Problem

A set of users U with $|U|$ members and a set of items V with $|V|$ items are considered. Each user $u \in U$ interacts with items, forming temporal sets. A temporal set s is defined as a subset of items $v \in V$, all sharing a common timestamp t. Each item v is represented as a one-hot encoded vector $\mathbf{v} \in \mathbb{R}^{|V|}$, and each temporal set s_i as a collection of such vectors $\mathbf{s}_i = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, where n denotes the number of items in s_i.

The interaction history of a user u is represented by a sequence of temporal sets $S^u = \{s_1^u, s_2^u, \ldots, s_L^u\}$, each with a corresponding timestamp t_i^u. The challenge lies in predicting the composition of future temporal sets based on this historical data.

$$\text{Predict } \hat{S}^u_{\text{next}} = \{s^u_{L+1}, s^u_{L+2}, \ldots, s^u_{L+k}\} \text{ given } S^u \tag{6.1}$$

Here, $\hat{S}^u_{\text{next}}$ denotes the predicted next k temporal sets for user u, with k being a predefined target sequence length. The prediction is based on the sequence S^u, encapsulating the user's interaction history with temporal sets with associated timestamps. The objective is to accurately forecast the items in these future sets, leveraging insights from the user's past interactions.

Example in Shopping Baskets: Consider a user u who has the following sequence of shopping baskets, each associated with a specific day:

$$S^u = \{\{\text{milk, bread}\}_{t_1}, \{\text{eggs, cheese}\}_{t_2}, \{\text{chicken, rice}\}_{t_3}\}$$

The task is to predict the next sequence of baskets $\hat{S}^u_{\text{next}}$, which might look like:

$$\hat{S}^u_{\text{next}} = \{\{\text{beef, pasta}\}_{t_4}, \{\text{apples, yogurt}\}_{t_5}\}$$

This prediction helps in providing personalized shopping recommendations to the user, enhancing their shopping experience.

Example in Website Interactions: On a content website, a user u interacts with various articles during different visits:

$$S^u = \{\{\text{Article A, Article B}\}_{t_1}, \{\text{Article C}\}_{t_2}, \{\text{Article D, Article E}\}_{t_3}\}$$

The goal is to predict the next set of articles $\hat{S}^u_{\text{next}}$ the user will interact with:

$$\hat{S}^u_{\text{next}} = \{\{\text{Article F, Article G}\}_{t_4}, \{\text{Article H, Article I}\}_{t_5}\}$$

This recommendation can improve the user's engagement by suggesting relevant content based on their reading history.

Example in Daily Purchases: For a daily deal website, a user u has the following sequence of daily purchases:

$$S^u = \{\{\text{Deal 1, Deal 2}\}_{t_1}, \{\text{Deal 3}\}_{t_2}, \{\text{Deal 4, Deal 5}\}_{t_3}\}$$

The task is to predict the next set of deals $\hat{S}^u_{\text{next}}$ the user is likely to purchase:

$$\hat{S}^u_{\text{next}} = \{\{\text{Deal 6, Deal 7}\}_{t_4}, \{\text{Deal 8, Deal 9}\}_{t_5}\}$$

By accurately predicting future purchases, the platform can tailor deals to the user's preferences, increasing satisfaction and sales.

6.3 Existing Shortcomings

In recent years, the domain of temporal sets recommendation has garnered significant attention, with researchers endeavoring to identify and implement effective inductive biases. A range of methodologies, such as Sets2Sets [28], DSNTSP [31], and DNNTSP [556], have been proposed, each offering unique perspectives and strategies to tackle this challenge. However, despite these advancements, existing approaches exhibit certain limitations that have yet to be fully addressed. These shortcomings, which are detailed in the subsequent paragraphs, underscore the need for continued research and development in this field.

Inadequate Modeling of Set Representation: Many studies [7, 26–28, 31, 556] rely on basic aggregation or pooling methods, such as max or mean operations, to derive a unified representation for set data. This simplistic approach often leads to significant information loss, especially concerning the relationships among items within and across sets. A more sophisticated strategy is required that adequately

considers both individual item representation and their interrelationships in forming the set representation.

Insufficient Modeling of Time Intervals: The majority of existing temporal sets recommendation models focus primarily on the order of interactions, neglecting the rich information provided by timestamps. Even in models that do incorporate timestamp data, the usage tends to be rudimentary. Relative Position Encoding Timestamp, for instance, can offer more specific insights into item relationships and user intentions than mere order sequencing. For example, the purchase of a computer followed by accessories like a mouse or keyboard within hours or days suggests a higher likelihood of relatedness compared to purchases made weeks or months apart. Pure order-based models fail to distinguish such nuances. [20] attempts to enhance single-item sequential recommendation by processing time intervals through a self-attention model, scaling them based on the shortest user-specific interval. However, this method underutilizes the potential of relative time information. A more potent approach is proposed, which utilizes advanced functions to quantify time intervals, mapping them to integers, and employing latent vectors for a more nuanced representation of relative time. Prior models have focused on utilizing absolute timestamp data, but it can be argued that representing relative positions is essential for modeling temporal sets. The wise utilization of time interval information, through appropriately chosen functions, can significantly enhance the modeling of these relationships. To fill in the gap mentioned in Section 1.2 of Handling Temporal Dynamics as well as well as Complexity in Item Set Recommendations, the Temporal Sets Nested Transformer (TSNT) is proposed, which effectively addresses these limitations, providing a comprehensive solution for temporal sets recommendation.

6.4 Contributions

The development and application of the Temporal Sets Nested Transformer (TSNT) provide multifaceted contributions, specifically addressing the fourth research question outlined in Section 1.2: *How to effectively model the temporal sets sequence with time interval information for recommending next temporal sets?*

- **Nested Transformer Architecture:** TSNT features a sophisticated nested transformer architecture comprising two key components: the Set Items Transformer (SIT) and the Temporal Sets Transformer (TST). The SIT, derived from the Transformer Encoder, encodes each set of items in a sequence, capturing distinct characteristics and intricate relationships among items, thus providing a granular understanding of the constituent items. The TST, operating at a higher level, integrates these detailed set representations over time, capturing sequence dynamics and effectively combining item-level details with temporal relationships between successive sets. This nested structure enables comprehensive and multi-dimensional encoding of sequence data, capturing both item-level details and temporal interconnections.

- **Novel Time Interval Processing:** A key innovation in TSNT is the Time Interval Quantizational Encoding (TIQE) method. This approach quantizes time intervals among sets and uses an embedding to map these quantized intervals into the model. By integrating this temporal information into the transformer, TSNT is able to understand and utilize the temporal dynamics present in user-item interaction data. This innovative method enhances the model's ability to discern temporal relationships between sets, providing a more nuanced representation of the temporal context.

- **Empirical Validation on Multiple Datasets:** TSNT has been tested across several renowned e-commerce datasets, including TaFeng, T-Mall, and Android Apps from Amazon Reviews. The model has demonstrated superior performance, outclassing existing benchmarks in terms of Hit Ratios and Normalized Discounted Cumulative Gains (NDCGs) at 10 and 20. These results highlight TSNT's ability to accurately predict future user interactions.

Overall, TSNT represents a significant advancement in temporal sets recommendation by effectively modeling the relationships within and between item sets while incorporating temporal dynamics. This innovative approach not only improves prediction accuracy but also enhances the understanding of complex user-item interactions over time.

6.5 Related Work

In this section, the related work on Temporal Sets Recommendation, Item Set Encoding Techniques, and Temporal Contextual Modeling is discussed. These areas represent the foundational research that has paved the way for the development of the Temporal Sets Nested Transformer (TSNT).

6.5.1 Temporal Sets Recommendation

Temporal sets recommendation leverages timestamped sequences of item sets to predict future sets a user is likely to interact with. Recent years have witnessed significant innovations in this domain. [28] introduced Sets2Sets, a recurrent neural network enhanced with a set attention mechanism and a repeated purchase mechanism for seq2seq prediction. [31] developed DSNTSP, which employs dual transformer networks on sequences of entities, integrated through a cross-transformer and a gating network. [556] proposed DNNTSP, combining graph convolutional networks with transformers to boost recommendation accuracy. These advancements underscore a trend towards incorporating sophisticated sequence modeling techniques with neural network architectures, leading to more precise predictions in temporal sets recommendation. Further exploration in this area includes the use of graph neural networks by [557], emphasizing the importance of capturing complex relationships within sets. [558] introduced a global-local attention mechanism to enhance the modeling of temporal dependencies in user-item interactions. Similarly, [559] proposed a deep temporal sets model that integrates temporal dynamics with deep learning techniques, improving the model's ability to capture intricate temporal patterns. [560] developed an element-wise attention mechanism to enhance the modeling of item relationships within sets, further improving recommendation accuracy. Moreover, [561] introduced a continuous-time attention mechanism to model temporal dynamics more effectively. Building on these advancements, the Temporal Sets Nested Transformer (TSNT) integrates detailed set representations and time interval data, significantly enhancing the model's ability to capture temporal relationships and provide accurate recommendations.

6.5.2 Item Set Encoding Techniques

Effective representation of item sets (baskets) [67, 255, 397, 414] is crucial for the performance of recommendation systems, particularly with temporal sets. Traditional encoding methods like one-hot encoding, while straightforward, often fail to capture complex relationships within sets. Embedding techniques, which map items to dense vectors, have become popular for encoding semantic similarities. More advanced approaches, such as graph-based methods, further enhance representation by considering structural relationships between items. [67] introduced models that account for complementarity, compatibility, and loyalty within shopping baskets, improving recommendation accuracy. [255] proposed neural basket embeddings capturing the sequential nature of user interactions. [397] utilized hypergraph convolutional networks to model complex dependencies within repeated baskets, while [414] highlighted the importance of incorporating temporal dynamics by adjusting item weights based on purchase intervals. The introduction of the Set Items Transformer (SIT) in TSNT builds on these advancements by creating rich, detailed set embeddings that capture the intricacies within each item set, thereby enhancing recommendation accuracy.

6.5.3 Temporal Contextual Modeling

Incorporating contextual information, particularly temporal context, has been shown to significantly enhance the personalization and relevance of recommendation systems. For instance, [169] introduced a contextualized temporal attention mechanism that integrates temporal dynamics with attention mechanisms, effectively capturing the contextual influence of historical user behaviors on current predictions. Building on this idea, [127] developed a model that combines multiple attention mechanisms with multi-temporal embeddings, enhancing predictive performance by capturing various temporal patterns in user behavior. Further emphasizing the importance of temporal information, [147] proposed a model that accounts for intricate temporal patterns to provide timely recommendations. In a similar vein, [267] explored the use of graph neural networks augmented with temporal information, capturing both structural and temporal aspects of user behavior. Extending this approach, [327] proposed a temporal graph transformer for multi-behavior sequential recommendation tasks, integrating various user behaviors and

their temporal dynamics. Additionally, [476] enriched temporal knowledge graphs with user behavior data, learning from both relational and temporal dimensions. To further advance this field, [353] presented a temporal graph contrastive learning approach, leveraging contrastive learning techniques to capture temporal dependencies in user-item interactions. The Temporal Sets Transformer (TST) within TSNT exemplifies these advancements by effectively assimilating and interpreting time interval data through Time Interval Quantizational Encoding (TIQE), providing a more nuanced representation of temporal relationships within sequences of item sets.

6.6 Methodology

In this section, the architecture of the Temporal Sets Nested Transformer (TSNT) (Figure 6.3), an innovative model designed for solving temporal sets recommendation problem is proposed. TSNT is adept at capturing and interpreting the temporal dynamics in sequences of item sets, providing nuanced and contextually relevant recommendations. The model consists of two key components: the Set Items Transformer (SIT) and the Temporal Sets Transformer (TST). The SIT encodes each set of items in a sequence, capturing the distinct characteristics of each set, including the intricate relationships among items. The TST integrates the detailed set representations over time, capturing the overarching sequence dynamics, effectively weaving together both the individual nuances of items within each set and the temporal relationships between successive sets. The model also introduces the Time Interval Quantizational Encoding (TIQE) method, an innovative approach to encoding time intervals that enhances the model's ability to discern temporal relationships between sets. Under the setup of encoder-decoder architecture, the model is trained to predict future sets of items for users based on their historical interactions. The detailed components and operations of TSNT are described in the following subsections.

6.6.1 Item Embedding

Each item v_i in a set within a sequence is represented through a one-hot encoded vector $\mathbf{v}_i$, which is then transformed into a d-dimensional latent space vector using

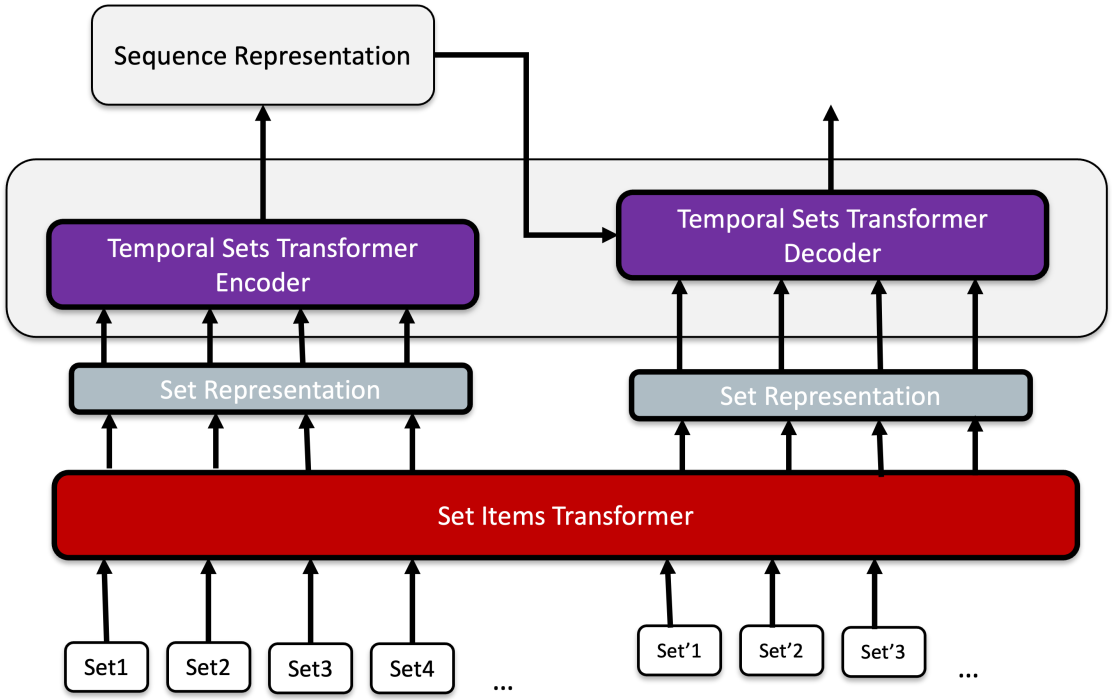


FIGURE 6.3: The Temporal Sets Nested Transformer (TSNT) Architecture

an embedding matrix $\mathbf{W}_x$:

$$x_i = \mathbf{W}_x \mathbf{v}_i \tag{6.2}$$

This process effectively captures the unique characteristics of each item in the latent space. By transforming the sparse one-hot encoded vector into a dense representation, the model gains the ability to discern subtler patterns and relationships among items, which is essential for understanding the nuances of each item's role and interaction within the set in the sequence.

6.6.2 Temporal Sets Nested Transformer (TSNT): Encoder

The encoder of TSNT is a nested transformer architecture, consisting of a Set Items Transformer (SIT) and a Temporal Sets Transformer (TST). This nested structure allows for a rich encoding of both the individual items in sets and the temporal relationships between sets.

Set Items Transformer. The Set Items Transformer (SIT) (Figure 6.4) encodes each set in a sequence, capturing the distinct characteristics of each set, including the intricate relationships among items. This component is crucial for generating

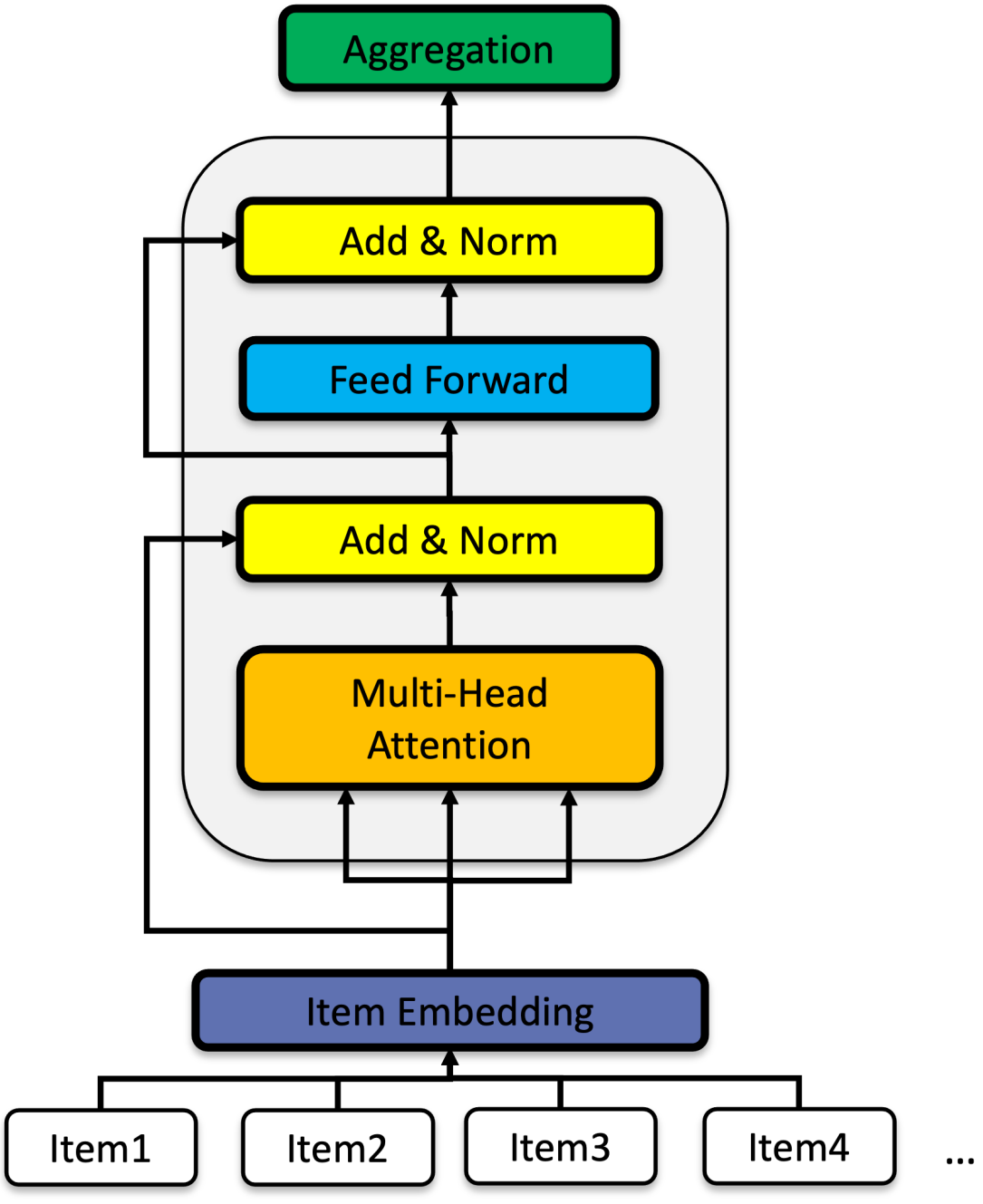


FIGURE 6.4: The Set Items Transformer (SIT) Module

rich, detailed set embeddings that capture the subtleties within each item set. These embeddings are then integrated into a higher-level transformer.

Set Items Input Vector The Set Items Transformer begins with a set of items $\mathbf{s}_0 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ as input, where $\mathbf{x}_i$ is the embedding of the i-th item in the set. 0 here means it is regarded as output of 0th layer. A maximum set size N is defined, and the vector is padded with zeros to ensure a consistent input size.

Multi-Head Self-Attention. Then, the multi-head self-attention mechanism is applied to capture the relationships between items within the set:

$$\mathbf{z}_j^{(l)} = \text{MultiHead}(\mathbf{s}_j, \mathbf{s}_j, \mathbf{s}_j) \tag{6.3}$$

for layer l, where $\mathbf{s}_j$ is the embedding of the j-th set with the 0th layer as the item input layer, and $\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ computes the attention for queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$. This mechanism allows the transformer to capture the intricate relationships between items in the set.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{6.4}$$

Here, Q, K, and V are the query, key, and value matrices, with respective weight matrices W_i^Q, W_i^K, and W_i^V for each head. W^O is the output weight matrix.

Add and Norm. Each sub-layer (including Multi-Head Attention) includes a residual connection followed by layer normalization:

$$\mathbf{x}_j^{(l)} = \text{LayerNorm}(\mathbf{s}_j + \mathbf{z}_j^{(l)}) \tag{6.5}$$

This step is crucial for stabilizing the learning process and enabling deeper architectures by mitigating the vanishing gradient problem.

Position-wise Feed Forward Net. After attention and normalization, a position-wise Feed Forward Network is applied:

$$\mathbf{x}_j^{(l)} = \text{LayerNorm}(\mathbf{x}_j^{(l)} + \text{FFN}(\mathbf{x}_j^{(l)})) \tag{6.6}$$

where $\text{FFN}(x) = \max(0, x\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$. The position here is unified for all items within the set using the set position in the sequence. And in the equation

above, another Add & Norm layer is applied to stabilize learning and maintain information flow after the Feed Forward Network.

Aggregated Set Embedding. The output of the Set Items Transformer is the aggregated set embedding, summed from the output of the last layer:

$$\mathbf{h}_j = \sum_{i=1}^{N} \mathbf{x}_i^{(L)} \tag{6.7}$$

This aggregated set embedding is then fed into the Temporal Sets Transformer for further processing with temporl and positional information.

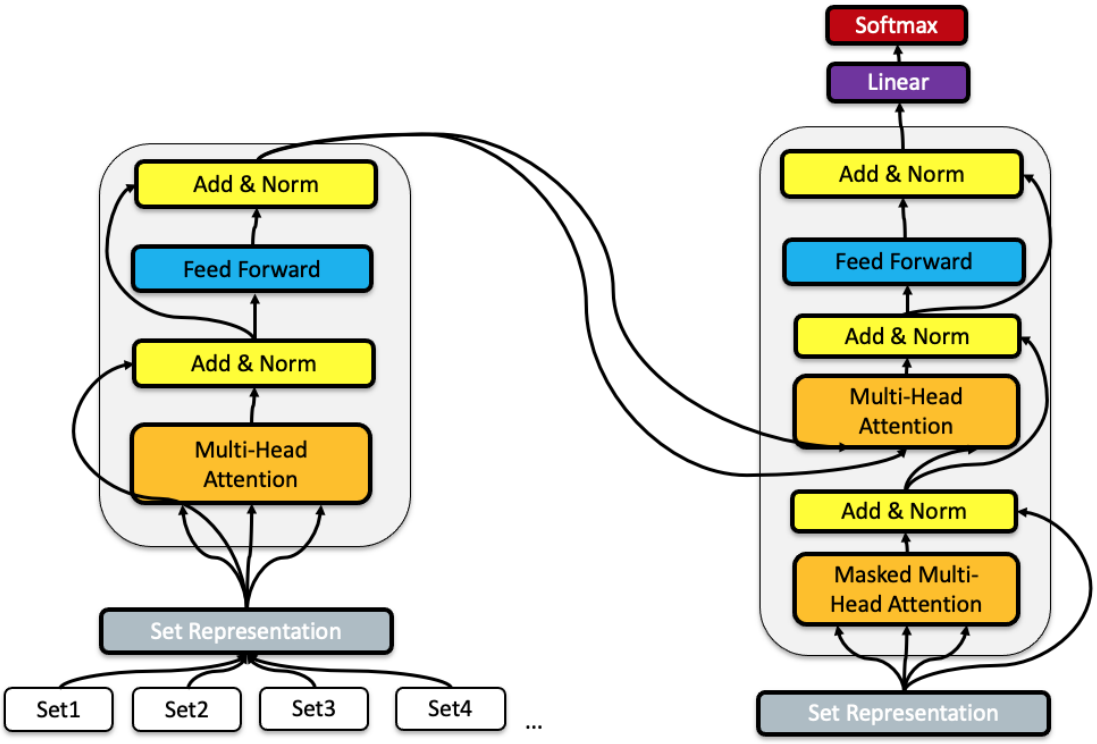


FIGURE 6.5: The Temporal Sets Transformer (TST) Module

Temporal Sets Transformer. The Temporal Sets Transformer (Figure 6.5) integrates the detailed set representations over time order within the sequence of sets, capturing the overarching sequence dynamics and effectively combining item-level details with temporal relationships between successive sets.

Set Positional Encoding. To encode the positional information of sets within the sequence, a modified sinusoidal positional encoding is used:

$$PE(pos, i) = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) & \text{if } i \mod 2 = 0 \\ \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) & \text{if } i \mod 2 = 1 \end{cases} \tag{6.8}$$

This encoding facilitates the model in understanding the sequential order of sets, providing critical information about the relative positions of different sets within the sequence.

Time Interval Quantizational Encoding (TIQE). The TIQE method is crucial for encoding temporal dynamics in TSNT. An enhanced formulation is proposed that allows for a more nuanced representation of time intervals. The new TIQE function is defined as follows:

$$TIQE(t_i, t_j) = f\left(\frac{t_i - t_j}{\Delta t}\right) \tag{6.9}$$

where t_i and t_j are the timestamps of the i-th and j-th sets, respectively, and Δt is the time interval quantization factor. The function $f(x)$ is a non-linear transformation applied to the normalized time difference. This non-linear transformation can be tailored based on experimental findings to best capture the temporal dynamics relevant to TSNT.

The quantized time interval is then mapped into a latent space using an embedding matrix $\mathbf{W}_t$:

$$\mathbf{t}_{ij} = \mathbf{W}_t \cdot TIQE(t_i, t_j) \tag{6.10}$$

This formulation introduces a layer of flexibility, allowing the model to adaptively learn the most effective way to represent time intervals based on the dataset and task at hand. The choice of $f(x)$ should be guided by empirical evidence, as different datasets and recommendation contexts might benefit from different temporal dynamics.

The selection of i and j determines the desired order of recency and introduces a hyperparameter for recency order. For instance, to utilize the last 3 sets, one can

choose i = 3, j = 2, and k = 1. Similarly, to use the last 5 sets, i = 5, j = 4, and k = 3 can be selected. This flexibility allows the model to adapt to different sequence lengths and recency requirements, enhancing its versatility and effectiveness across various recommendation scenarios.

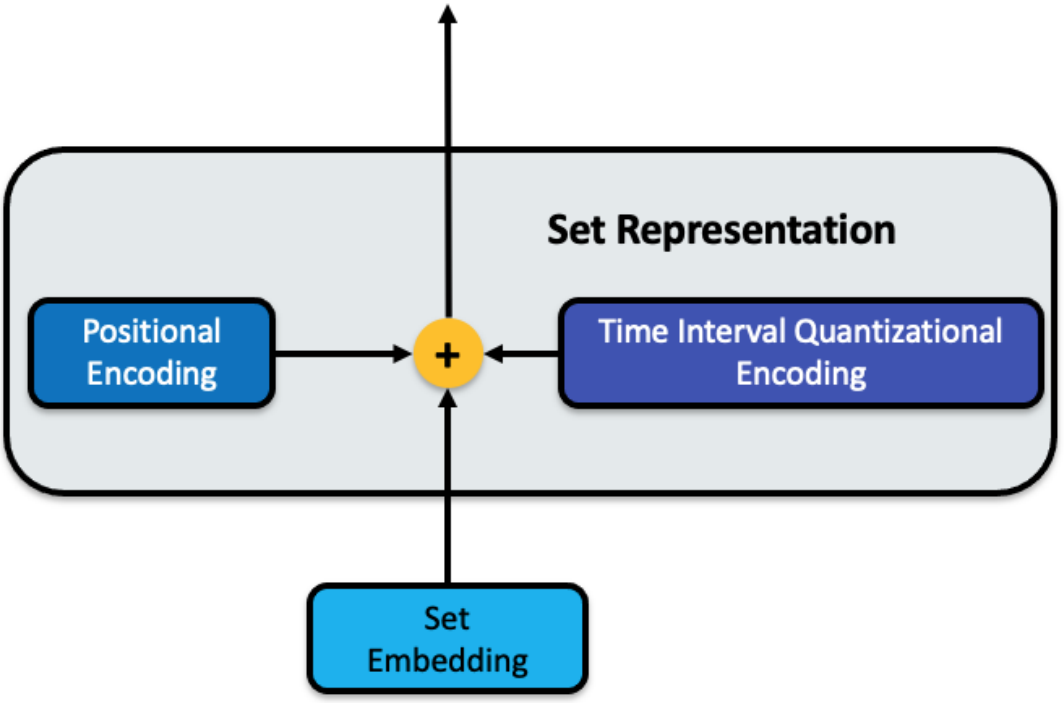


FIGURE 6.6: The Set Representation Module

Set Representation. Then the set embedding obtained from the Set Items Transformer is concatenated with the PE and TIQE embeddings to form the final set representation (Figure 6.6):

$$\mathbf{h}_j = \mathbf{h}_j + PE(j) + \mathbf{t}_{ij} \tag{6.11}$$

This final set representation is then fed into the Temporal Sets Transformer for further processing.

Sequence Set Input Vector. The final set embeddings are form the input sequence for the Temporal Sets Transformer are the vectors obtained from the Set Items Transformer, positional encoding, and time interval quantization encoding:

$$\mathbf{H}_0 = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L\} \tag{6.12}$$

where $\mathbf{h}_j$ represents the j-th set in the sequence. $\mathbf{H}_0$ denotes the set of all set representations in the sequence, with the subscript 0 indicating it is the output of the initial (0th) layer. A maximum sequence length L is defined, and vectors are padded with zeros to ensure a consistent input size.

Multi-Head Self-Attention. The Temporal Sets Transformer employs a Multi-Head Attention mechanism to capture the intricate relationships across different sets in the sequence:

$$\mathbf{z}_j^{(l)} = \text{MultiHead}(\mathbf{h}_j, \mathbf{h}_j, \mathbf{h}_j) \tag{6.13}$$

for layer l, where $\mathbf{s}_j$ is the embedding of the j-th set with the 0th layer as the item input layer, and $\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ computes the attention for queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$. This mechanism allows the transformer to capture the intricate relationships between items in the set.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{6.14}$$

Here, Q, K, and V are the query, key, and value matrices, with respective weight matrices W_i^Q, W_i^K, and W_i^V for each head. W^O is the output weight matrix.

Add and Norm. Following the attention mechanism, an Add & Norm layer is applied, featuring a residual connection and layer normalization:

$$\mathbf{x}_j^{(l)} = \text{LayerNorm}(\mathbf{h}_j + \mathbf{z}_j^{(l)}) \tag{6.15}$$

This helps to maintain information throughout the layers and stabilize learning.

Position-wise Feed Forward Net. Each set representation is further processed through a position-wise Feed Forward Network:

$$\mathbf{x}_j^{(l)} = \text{LayerNorm}(\mathbf{x}_j^{(l)} + \text{FFN}(\mathbf{x}_j^{(l)})) \tag{6.16}$$

where $\text{FFN}(x) = \max(0, x\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$. This network introduces additional non-linearity and complexity into the model, allowing it to capture more intricate temporal patterns. Another Add & Norm layer is applied to stabilize learning and maintain information flow after the Feed Forward Network.

Aggregated Sequence Embedding. The output of the Temporal Sets Transformer is the aggregated sequence embedding, summed from the output of the last

layer:

$$\mathbf{X} = \sum_{j=1}^{L} \mathbf{x}_j^{(L)} \tag{6.17}$$

This aggregated sequence embedding is then fed into the decoder for further processing.

6.6.3 Temporal Sets Nested Transformer (TSNT): Decoder

The decoder in the Temporal Sets Nested Transformer (TSNT) model parallels its encoder, employing a similar nested transformer architecture to generate the final set of recommendations. It uses the encoded sequence representations to predict future user interactions, focusing on temporal dynamics and interaction history.

Set Items Transformer Decoder. The Set Items Transformer Decoder processes each set's representation in the sequence, utilizing the encoded information to predict the next items. This component is crucial for generating accurate item-level predictions based on the detailed embeddings from the encoder. It follows the same structure as the Set Items Transformer in the encoder, so the details are omitted here.

Temporal Sets Transformer Decoder. The Temporal Sets Transformer Decoder integrates the detailed set representations over time, capturing the overarching sequence dynamics and effectively combining both the individual nuances of items within each set and the temporal relationships between successive sets. Besides the Masked Multi-Head Self-Attention and the Prediction and Recommendation, the decoder also follows the same structure as the Temporal Sets Transformer in the encoder. The details flow can be referenced in Figure 6.5.

Masked Multi-Head Self-Attention. The multi-head self-attention mechanism with masking is applied to capture the relationships between items within the predicted set while preventing information flow from future tokens:

$$\mathbf{z}_j^{(l)} = \text{MultiHead}(\mathbf{s}_j, \mathbf{s}_j, \mathbf{s}_j, \text{mask}) \tag{6.18}$$

for layer l, where $\mathbf{s}_j$ is the embedding of the j-th set, and the mask ensures that predictions are made based only on known past and present items.

Prediction and Recommendation. Central to the decoder is the Prediction Layer, designed to forecast the next set of recommendations. This layer synthesizes the encoded temporal context and user-item interaction patterns into actionable predictions:

$$\text{score} = \text{softmax}(\mathbf{W}_{\text{pred}} \cdot \mathbf{h}_{\text{last}} + \mathbf{b}_{\text{pred}}) \tag{6.19}$$

Here, $\mathbf{h}_{\text{last}}$ is the last hidden state from the decoder, incorporating the temporal and interaction dynamics. $\mathbf{W}_{\text{pred}}$ and $\mathbf{b}_{\text{pred}}$ are the prediction layer's weight and bias.

This comprehensive decoding process ensures that the model effectively captures and utilizes the temporal dynamics and interaction history to generate accurate next-item recommendations.

The TSNT model predicts future user behavior by analyzing the last L interaction sets, characterized by both items and their timestamps. The model's output layer, y, computes softmax scores for each item, indicating their likelihood as the next preferred choice. This probability is calculated considering both the historical interactions and their temporal order.

Recommendations are then generated by selecting the top N items with the highest softmax scores for each step, ensuring that the suggestions are not only based on past preferences but also relevant to the current temporal context. This dynamic approach allows TSNT to provide accurate, contextually relevant recommendations that mirror real-world user behavior and preferences.

6.6.4 Model Training

The TSNT model is trained using the Binary Cross-Entropy (BCE) loss function, which measures the difference between predicted probabilities and actual binary outcomes. For a predicted probability $\hat{y} \in [0, 1]$ and the actual label $y \in \{0, 1\}$, the BCE loss is calculated as:

$$\mathcal{L}_{\text{BCE}} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]. \tag{6.20}$$

where $y = 1$ indicates a positive interaction, while $y = 0$ indicates a negative interaction. A 3:1 negative sampling ratio is adopted, following [1, 2, 11].

In the encoder-decoder framework, BCE is crucial for predicting user-item interactions, helping the model distinguish between potential interaction (label 1) and non-interaction (label 0) scenarios. This choice is motivated by BCE's capability to handle the typically imbalanced nature of user-item interaction data in recommendation systems. The integration of BCE in the model ensures refined tuning of parameters, allowing for more accurate predictions of future user interactions.

6.7 Experiment Setup

In this section, experimental setup is described, including datasets, evaluation metrics, and baselines.

6.7.1 Datasets

TABLE 6.1: Statistics of the datasets

Dataset	#users	#items	#baskets	#interactions	sparsity	#categories	max basket size	avg #items per basket
TaFeng	4419	10817	54897	714893	98.50%	1357	109	13.02
T-Mall	5257	13202	80294	143058	99.79%	70	20	1.78
Android Apps	4871	7191	42998	86989	99.75%	Nil	58	2.02

This study employs datasets in Table 6.1 from three distinct sources: **TaFeng**, **T-Mall**, and **Android Apps** from Amazon Reviews, each offering a unique perspective on consumer purchasing behaviors.

TaFeng Dataset [1] encompasses four months of comprehensive basket purchase data from a grocery store, providing insights into consumer buying patterns in a supermarket context. It is suitable for temporal sets recommendation as it captures frequent and varied purchases. Each day's purchases can be aggregated into a set, forming a sequence of sets over time.

T-Mall Dataset [2], released by the Alibaba Group, captures the vibrancy of e-commerce interactions in China, reflecting a diverse range of consumer choices on a large online shopping platform. Its large scale and variety make it an excellent

[1] `http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng`
[2] `https://tianchi.aliyun.com/dataset/dataDetail?dataId=53&userId=1`

source for studying temporal sets recommendation in an online retail setting. Purchases made on the same day can be aggregated into a set to form sequences of sets.

Amazon Reviews Dataset [3], specifically the **Android Apps** section, offers a deep dive into user-item interactions on Amazon's platform. Although the primary focus is on the interactions rather than the review content, it provides valuable insights into user preferences and behaviors [651–653]. It is suitable for temporal sets recommendation studies where users often purchase multiple related items on the same day.

Preprocessing Steps: The datasets undergo several preprocessing procedures to ensure consistency and relevance:

- Establishing a chronological order for user-item interactions based on timestamps.
- Filtering out users with less than **10** items and items with fewer than **10** users [1, 11, 651, 652, 654, 655]. Post aggregation into set sequences, users with fewer than **7** sets were also excluded to ensure a reasonable sequence length for each user. This approach primarily targets warm-start scenarios, aiding in mitigating cold-start issues.
- Utilizing only those interactions from the Amazon dataset where the rating is **3** or higher, as this threshold is indicative of user preference. Given the absence of explicit sets in Amazon data, the dataset is aggregated by considering one day as the time frame for each set and treated items reviewed within a day as a single set.

Those datasets are selected mainly due to their diverse characteristics and the unique insights they offer into user-item interactions to form suitable temporal sets data. The TaFeng dataset provides a comprehensive view of consumer purchasing patterns in a supermarket setting, while the T-Mall dataset captures the vibrancy of e-commerce interactions on a large online platform. The Amazon Reviews dataset, specifically the Android Apps section, offers a deep dive into user-item interactions on Amazon's platform, providing valuable insights into user

[3] `http://jmcauley.ucsd.edu/data/amazon/`

preferences and behaviors. The combination of these datasets allows for a comprehensive evaluation of the Temporal Sets Nested Transformer (TSNT) model across different recommendation scenarios and user-item interaction contexts.

6.7.2 Baselines

To evaluate the performance of the Temporal Sets Nested Transformer (TSNT), it is compared with the following 7 baselines available at the time this research was conducted. These baselines include both classical and state-of-the-art models from traditional methods to deep learning approaches, covering a wide range of recommendation techniques.

PopRec: This baseline recommends items based purely on their popularity, measured by the frequency of each item's occurrence across the entire dataset. It serves as a non-personalized benchmark. For the temporal sets recommendation, the popularity is calculated based on the most popular items within the user's history sets.

PersonalPopRec: A personalized version of the popularity-based recommendation, it considers the most popular items within a user's history. This baseline is designed to capture user-specific preferences while maintaining the simplicity of the popularity-based approach.

kNN: The k-Nearest Neighbors algorithm is a collaborative filtering method that recommends items based on the preferences of similar users. The most nearest neighbors are selected based on the cosine similarity between users' history sets for termporl sets recommendation.

FPMC [1]: FPMC blends Matrix Factorization (MF) with First-Order Markov Chains (FMC) to model sequential patterns and general user preferences. For a seq2seq setup, the model predicts the next set of items using the last predicted sets of items as one more set for the input sequence.

DREAM [7]: It uses Recurrent Neural Networks (RNN) to capture the sequential signals in user's interaction history. Under the seq2seq setup, the model predicts the next set of items based on the last set of items predicted as one more set for the input sequence.

Sets2Sets [28]: This model is designed for sequence prediction, treating each sequence as a set and applying a set-to-set translation mechanism. It is adapted for temporal sets recommendation by predicting the next set of items based on the last set of items predicted as one more set for the input sequence.

DSNTSP [31]: This work is a state-of-the-art model for temporal sets recommendation at the time when the study was conducted, leveraging a dual sequential network to capture both the sequential and temporal dynamics of user-item interactions. The model predicts the next set of items based on the last set of items predicted as one more set for the input sequence.

These baselines were selected to provide a comprehensive evaluation of the TSNT model, covering a wide range of recommendation techniques from traditional methods to state-of-the-art deep learning approaches. The comparison with these baselines allows for a thorough assessment of the TSNT model's performance across different recommendation scenarios and user-item interaction contexts.

6.7.3 Evaluation Protocols

Evaluating the effectiveness of the model is crucial for temporal sets recommendation. The performance of the Temporal Sets Nested Transformer (TSNT) model is evaluated using two key metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), with cut-off points at 10 and 20. Since the setup is a seq2seq model, the final evaluation results are averaged over the target length.

- **Hit Ratio@K (HR@K)**: Measures whether the actual items a user interacted with appears in the top-K recommendations for the set. A higher HR@K indicates better performance in capturing recent user preferences.
- **NDCG@K (Normalized Discount Cumulative Gain)**: A position-aware metric assigning larger weights to items ranked higher. Higher values indicate more effective ranking of items by their recent relevance.

6.7.4 Implementations

The Temporal Sets Nested Transformer (TSNT) and baseline models were implemented using PyTorch [656], incorporating elements from the original authors' versions as necessary. Experiments were conducted on Nvidia V100 and RTX 2080ti GPUs. Hyperparameters were initially optimized via grid search and then kept constant for subsequent runs. The results on the testing set, corresponding to the best validation metrics, were averaged over the 10 runs and reported. Early stopping was employed during training to halt when validation loss and metrics plateaued.

In terms of hyperparameter tuning, the optimizer for all models was Adam [631], selected for its efficiency. The learning rate was tuned within the set {1e-3, 3e-4, 1e-4}, and ultimately set at 1e-3. L2 regularization was adjusted within the range {1e-1, 1e-2, ..., 1e-6} and fixed at 1e-6. The latent dimension, crucial for model performance, was tested across {16, 32, 64, 128, 256} and finalized at 128. The batch size, a key factor in model training, was set at 256, chosen from {16, 32, 64, 128, 256}. Dropout [657] was employed to enhance model generalization, with values tuned between {0.2, 0.5}. Model parameters were initialized using Xavier initialization [659] or according to the original model settings, depending on the specific requirements.

For TSNT, The embedding size was set to 64, the number of attention heads to 8, and the number of layers to 2. The time interval quantization factor was set to 1 with Exponential Decay function. For TSNT-specific configurations for both two nested transformers, the number of blocks was a crucial parameter, tested within the range {1, 2}. The number of layers in the MLP component was similarly varied within {1, 2} to optimize performance. DREAM's layer count was crucial for its performance and was tuned from {1, 2, 3, 4}. As for Sets2Sets, the number of layers was tested within {1, 2, 3}. The number of layers in the DSNTSP model was optimized within {1, 2, 3} to enhance its performance. By tuning these hyperparameters, the models were optimized for performance and efficiency, ensuring that the results accurately reflect their capabilities and effectiveness in the temporal sets recommendation context.

6.8 Experiment Results and Analysis

This study is guided by the following key research questions, each addressing a distinct aspect of the Temporal Sets Nested Transformer (TSNT):

RQ1: How does TSNT perform relative to other recommendation models?

RQ2: What contribution does each component of TSNT make to its overall effectiveness?

RQ3: Among the TIQE functions, which one enhances the TSNT's performance the most? What is the best choice of δt?

These research questions aim to provide a comprehensive understanding of TSNT, from its basic performance and internal mechanisms to its scalability, efficiency, and ability to provide insights into user behavior.

6.8.1 Recommendation Performance

The performance of TSNT was evaluated against seven baseline recommendation systems. Each model was run 10 times, and the average results of these runs are presented in Table 6.2.

TABLE 6.2: Performance of Baseline Models and TSNT

Dataset	Metric	PR	PPR	kNN	FPMC	DRM	S2S	DSNTSP	TSNT	Rel Gain
TaFeng	HR@10	0.1234	0.1523	0.1432	0.1711	0.2012	0.2145	0.2310	**0.2352**	1.82%
	NDCG@10	0.0832	0.1107	0.0921	0.1234	0.1543	0.1602	0.1823	**0.1891**	3.73%
	HR@20	0.2034	0.2251	0.2133	0.2502	0.2845	0.2954	0.3107	**0.3223**	3.74%
	NDCG@20	0.1431	0.1657	0.1572	0.1983	0.2254	0.2356	0.2578	**0.2634**	2.17%
T-Mall	HR@10	0.1321	0.1584	0.1476	0.1763	0.2057	0.2189	0.2364	**0.2447**	3.52%
	NDCG@10	0.0884	0.1156	0.0997	0.1305	0.1619	0.1687	0.1901	**0.1965**	3.37%
	HR@20	0.2086	0.2297	0.2189	0.2558	0.2892	0.3003	0.3156	**0.3286**	4.12%
	NDCG@20	0.1487	0.1712	0.1623	0.2029	0.2301	0.2403	0.2625	**0.2710**	3.24%
Android Apps	HR@10	0.1357	0.1609	0.1498	0.1785	0.2078	0.2210	0.2385	**0.2459**	3.10%
	NDCG@10	0.0903	0.1175	0.1016	0.1324	0.1638	0.1706	0.1920	**0.1992**	3.75%
	HR@20	0.2129	0.2340	0.2232	0.2601	0.2935	0.3046	0.3199	**0.3319**	3.75%
	NDCG@20	0.1509	0.1734	0.1645	0.2051	0.2323	0.2425	0.2647	**0.2731**	3.18%

The model abbreviations are defined as: PR (PopRec), PPR (PersonalPopRec), kNN (k-Nearest Neighbors), FPMC (Factorizing Personalized Markov Chains), DRM (DREAM), S2S (Sets2Sets), DSNTSP (Dual Sequential Network for Temporal Sets Prediction), and TSNT (Temporal Sets Nested Transformer).

The remarkable performance of the Temporal Sets Nested Transformer (TSNT) model across diverse datasets and metrics underscores its proficiency in the temporal sets recommendation task. Consistently outperforming baseline models such as PopRec, PersonalPopRec, kNN, FPMC, DREAM, Sets2Sets, and DSNTSP, TSNT demonstrates its robustness in capturing user preferences and predicting relevant sets of items. In the TaFeng dataset, characterized by diverse product ranges and routine purchasing patterns, TSNT achieves the highest Hit Rate (HR@10) of 0.2352 and NDCG@10 of 0.1891, with relative gains of 1.82% and 3.73% over the best baseline. Similarly, in the T-Mall dataset, reflecting intricate consumer behavior in e-commerce, TSNT leads with HR@10 of 0.2447 and NDCG@10 of 0.1965, showing relative gains of 3.52% and 3.37%. For the Android Apps dataset, involving sparse user-item interactions, TSNT maintains its superior performance with HR@10 of 0.2459 and NDCG@10 of 0.1992, achieving relative gains of 3.10% and 3.75%. TSNT's improvements in HR@20 and NDCG@20 metrics across all datasets further support its capability to predict both immediate and future purchases effectively, reaffirming its adaptability and efficiency in handling diverse data characteristics. Overall, TSNT's consistent outperformance across different datasets and metrics underscores its versatility and effectiveness in capturing temporal dynamics and user preferences, making it a powerful tool for recommendation tasks in various domains.

6.8.2 Ablation Study

To dissect the individual contributions of each component in TSNT, an ablation study is conducted. This study involved omitting each component from the full TSNT model and evaluating the impact on its performance. The results are summarized in Table 6.3.

Table 6.3: Ablation Study Results for TSNT on TaFeng Dataset

Model Variant	**NDCG@10**	**Improvement**
TSNT (Full Model)	0.432	-
w/o Item Embeddings	0.390	-9.72%
w/o Set Items Transformer	0.410	-5.09%
w/o Positional Encoding	0.420	-2.78%
w/o Time Interval Quantizational Encoding	0.398	-7.87%

The ablation study results provide critical insights into the importance of each component within the TSNT model. The full TSNT model achieves an NDCG@10 of 0.432 on the TaFeng dataset. When item embeddings are omitted, the model's performance drops to an NDCG@10 of 0.390, reflecting a significant decrease of 9.72%, indicating that item embeddings are crucial for capturing item-specific features and their interactions. Similarly, removing the set items transformer results in a performance decrease to an NDCG@10 of 0.410, a 5.09% reduction, highlighting its importance in modeling relationships among items within a set. The positional encoding component, essential for capturing the order of items, causes a 2.78% performance drop (NDCG@10 of 0.420) when omitted. The time interval quantizational encoding, which captures temporal dynamics between user interactions, results in an NDCG@10 of 0.398, translating to a 7.87% decrease in performance, underscoring its role in enhancing the model's ability to understand and leverage the timing of interactions. Overall, the ablation study demonstrates that each component of the TSNT model significantly contributes to its performance, with item embeddings and the set items transformer being particularly influential, while positional encoding and time interval quantizational encoding also play crucial roles. These findings underscore the efficacy of the nested transformer architecture in capturing temporal dynamics and user preferences, ultimately enhancing the predictive accuracy of TSNT.

6.8.3 Effectiveness of TIQE

The selection of an appropriate Time Interval Quantizational Encoding (TIQE) function is crucial for the effectiveness of the Temporal Sets Nested Transformer. To understand the impact of different TIQE functions on the performance of TSNT, various decay functions were experimented, each evaluated at different Δt values. This section discusses the findings, highlighting the TIQE function that most significantly enhances the model's performance and exploring the underlying reasons for this effectiveness.

The results presented in Table 6.4 indicates that the Exponential Decay function with a Δt of 0.5 achieves the highest NDCG@10 score (0.435), setting a benchmark for comparison. When the Δt is increased to 1.0 and 2.0, the effectiveness of this function decreases slightly, demonstrating a -0.69% and -1.38% drop in

TABLE 6.4: Performance of Different TIQE Functions in TSNT on TaFeng Dataset with Varying Δt

TIQE Function	Δt	**NDCG@10**	**Relative Improvement**
Exponential Decay	0.5	0.435	-
Exponential Decay	1.0	0.432	-0.69%
Exponential Decay	2.0	0.429	-1.38%
Linear Decay	0.5	0.423	-2.76%
Linear Decay	1.0	0.420	-3.45%
Linear Decay	2.0	0.417	-4.14%
Polynomial Decay	0.5	0.431	-0.92%
Polynomial Decay	1.0	0.428	-1.61%
Polynomial Decay	2.0	0.425	-2.30%
Logarithmic Decay	0.5	0.418	-3.91%
Logarithmic Decay	1.0	0.415	-4.60%
Logarithmic Decay	2.0	0.412	-5.29%

performance, respectively. This trend is observed across other TIQE functions as well, with each showing a decrease in performance as Δt increases. Notably, the Exponential Decay function consistently outperforms Linear, Polynomial, and Logarithmic Decays at all Δt values, underscoring its superior capability in capturing the temporal dynamics essential for accurate recommendations. This suggests that the Exponential Decay function, with its effective emphasis on more recent interactions, aligns closely with user behavior patterns, thus enhancing TSNT's recommendation accuracy.

6.9 Summary

This chapter introduced the **Temporal Sets Nested Transformer (TSNT)** model, a neural inductive bias designed to address the complex challenge of temporal sets recommendation. By tackling the fourth research question—*How to effectively model the temporal sets sequence with time interval information for recommending next temporal sets?*—TSNT advances the state of the art in modeling user interactions that consist of time-indexed sets of items. At the heart of TSNT is a nested transformer architecture comprising two key components: the **Set Items Transformer (SIT)** which encodes each item set in the sequence by capturing both item-level details and relationships among items within a set through

multi-head self-attention, layer normalization, and position-wise feed-forward layers for effective intra-set dynamics modeling; and the **Temporal Sets Transformer (TST)** which builds on the SIT outputs to process the entire sequence of sets by incorporating *Positional Encoding* for sets' order information and integrating the proposed *Time Interval Quantizational Encoding* (**TIQE**) to encode time intervals among sets via quantization, enriching the model with relative temporal context beyond simple sequential order. Through extensive experiments on multiple real-world e-commerce datasets, TSNT consistently outperforms strong baseline models, demonstrating higher Hit Ratios and NDCG scores at common cutoffs, while ablation studies confirm the importance of both components in the nested transformer design and the critical role of TIQE in capturing nuanced temporal signals—ultimately offering a powerful framework for accurately predicting future temporal item sets and enhancing recommendation quality across diverse application domains.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis advances the field of sequential recommendation by addressing three core challenges centered around the central research question: **How can neural inductive biases be devised to enhance sequential recommendation systems by effectively modeling variable interaction lengths, complex item relationships, and temporal dynamics?** A comprehensive literature review in Chapter 2 established the theoretical foundation and identified critical gaps in existing approaches, while highlighting the potential of carefully designed inductive biases. Building on these insights, this work introduced four novel neural inductive biases shown as below that effectively bridge the critical gaps outlined in the central research question and Section 1.2. Figure 7.1 illustrates how each proposed solution systematically addresses the three core challenges through four distinct sequential recommendation task setups.

The key contributions of 4 proposed neural inductive biases are summarized below:

- **Multi-Scale Quasi-RNN (QR-Rec):** This model addresses the challenge of modeling variable interaction lengths by introducing a multi-scale convolution mechanism within a recurrent neural network framework. Traditional approaches often rely on fixed-length windows or single-scale architectures, which can miss important features at different level of sub-sequences. QR-Rec overcomes this limitation through its hierarchical structure that processes

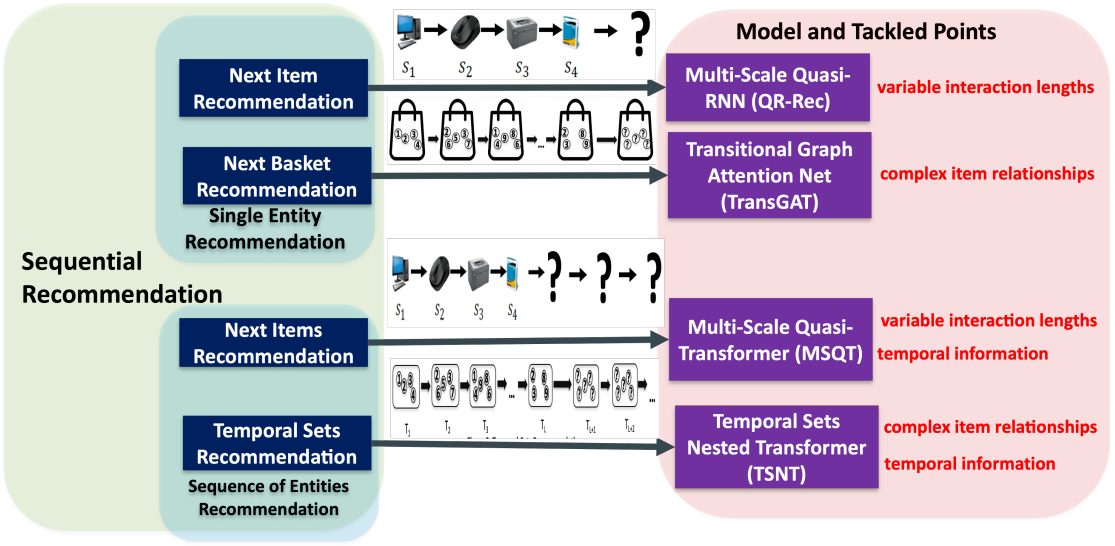


FIGURE 7.1: Summary of Sequential Recommendation Tasks and Solutions

user-item interactions at multiple temporal scales simultaneously, enabling more comprehensive sequence modeling and improving next-item recommendation accuracy.

- **Transitional Graph Attention Net (TransGAT):** This model tackles the challenge of modeling complex item relationships, particularly in next-basket recommendation scenarios where understanding transitions between items is crucial. While previous approaches often treated items independently or used simplified relationship models, TransGAT leverages graph attention mechanisms to explicitly capture and learn the intricate transitional patterns between items across consecutive baskets. The model's ability to differentiate and weight various types of item-item relationships leads to more nuanced and accurate basket-level predictions.

- **Multi-Scale Quasi-Transformer (MSQT):** This architecture addresses both the variable interaction length and temporal dynamics challenges through an innovative integration of transformer mechanisms with multi-scale convolution and temporal dynamic warping. Unlike conventional transformer-based approaches that may struggle with varying sequence lengths and temporal irregularities, MSQT's multi-scale design enables it to capture more comprehensive subsequence features. The temporal dynamic warping component specifically addresses the challenge of modeling irregular time intervals between interactions.

- **Temporal Sets Nested Transformer (TSNT):** This model simultaneously addresses the challenges of modeling complex item relationships and temporal dynamics in the context of temporal set sequence prediction. Traditional approaches often struggle to capture both the internal structure of item sets and their temporal evolution. TSNT's nested transformer architecture explicitly models both intra-set and inter-set relationships, while its temporal interval quantization encoding method effectively captures the temporal aspects of user behavior patterns. This dual focus enables more accurate modeling of both the structural and temporal dimensions of user-item interactions.

7.2 Future Work

Building on the insights and advancements presented in this thesis, several promising directions for future research in sequential recommendation systems are proposed below:

7.2.1 A Unified Sequential Recommendation Framework for Multiple Tasks

As shown in Figure 7.2, although separate models for each sequential recommendation task have proven successful, they also indicate the potential for a more unified approach. The four primary tasks—next item, next basket, next item sequence, and next temporal set sequence recommendations—share underlying sequential properties but differ in their input/output formats and target applications. Future research can thus focus on integrating these subtleties into a unified framework that not only maintains high performance but also streamlines system implementation.

However, achieving this unification demands solutions to multiple challenges, particularly those related to heterogeneous data representations and varying temporal dynamics, as illustrated in Figure 7.2. Designing adaptable architectures capable of handling these differences while consistently delivering strong results will be essential. A single framework that accommodates diverse task requirements could not only simplify deployment but also advance our understanding of the shared

principles behind different recommendation scenarios, paving the way for novel, cross-cutting approaches.

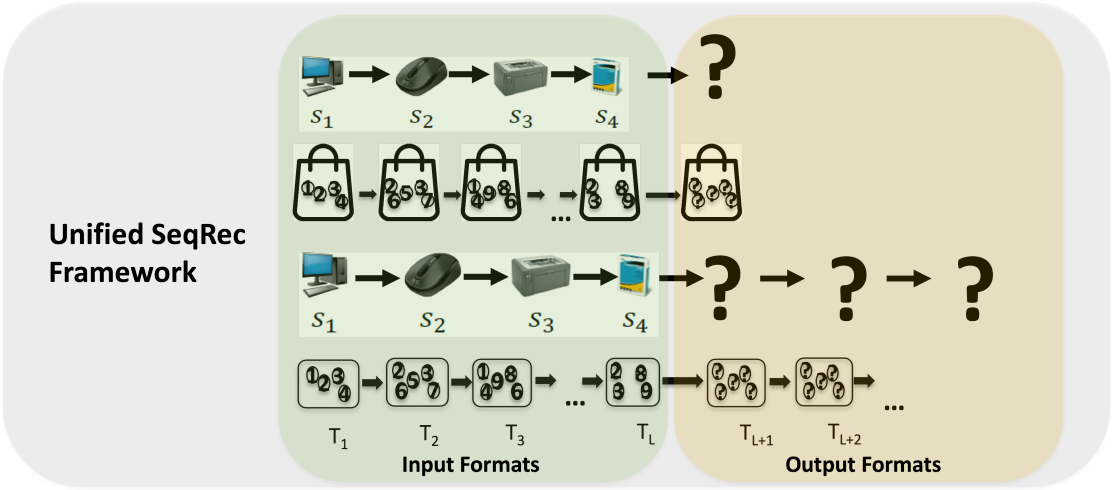


FIGURE 7.2: A unified sequential recommendation framework for multiple tasks.

7.2.2 Integrating Large Language Models into Sequential Recommendation Systems

As depicted in Figure 7.3, recent developments in Large Language Models (LLMs) offer compelling avenues for enhancing sequential recommendation systems. Future work could explore integration strategies centered around three main areas: Retrieval-Augmented Generation (RAG) [660] and Reinforcement Learning with Human Feedback (RLHF) [661]. RAG can exploit extensive external knowledge sources to improve contextual awareness, while RLHF enables refined alignment with user preferences, ultimately delivering more relevant and satisfying recommendations.

Additionally, investigating a variety of fine-tuning techniques [662–664] could adapt these large models to specific recommendation tasks while retaining their generalization capabilities. Methods such as prompt engineering, parameter-efficient fine-tuning, and domain-specific pre-training could prove invaluable in leveraging the semantic depth and generative strengths of LLMs without compromising speed or accuracy. This approach has the potential to expand the frontiers of sequential recommendation by enhancing both personalization and system adaptability.

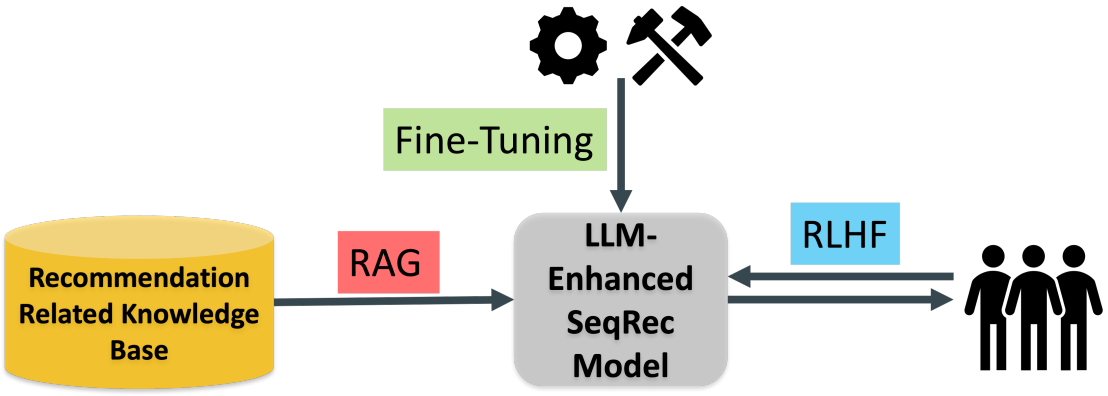


FIGURE 7.3: Integrating Large Language Models into Sequential Recommendation Systems.

7.2.3 Exploring SeqRec-Inspired Neural Inductive Biases for Efficient Language Models

As shown in Figure 7.4, while transformers have dominated the backbone of Large Language Models (LLMs), their substantial computational demands present significant hurdles for large-scale deployment. The success of alternative architectures in sequential recommendation—such as Quasi-RNNs [51] and Graph Neural Networks [53]—suggests promising routes for creating more efficient language models. Recent advances like the state-space based Mamba [54] and RNN-inspired RWKV [665] architectures demonstrate the viability of non-transformer approaches. These innovations indicate that investigating how sequential recommendation architectures might be adapted to language modeling could help maintain or even improve performance while reducing the computational overhead commonly associated with transformers.

Such studies could examine hybrid architectures that merge the efficient sequential processing capabilities of Quasi-RNNs or the structured reasoning of graph-based models with the representational power of transformers. As illustrated in Figure 7.4, the result might be language models that are not only more resource-efficient but also preserve or enhance the strong linguistic capabilities of current LLMs through novel architectural combinations. These advances would contribute significantly to both the specialized realm of sequential recommendation and the broader pursuit of scalable, accessible, and environmentally sustainable large-scale language models.

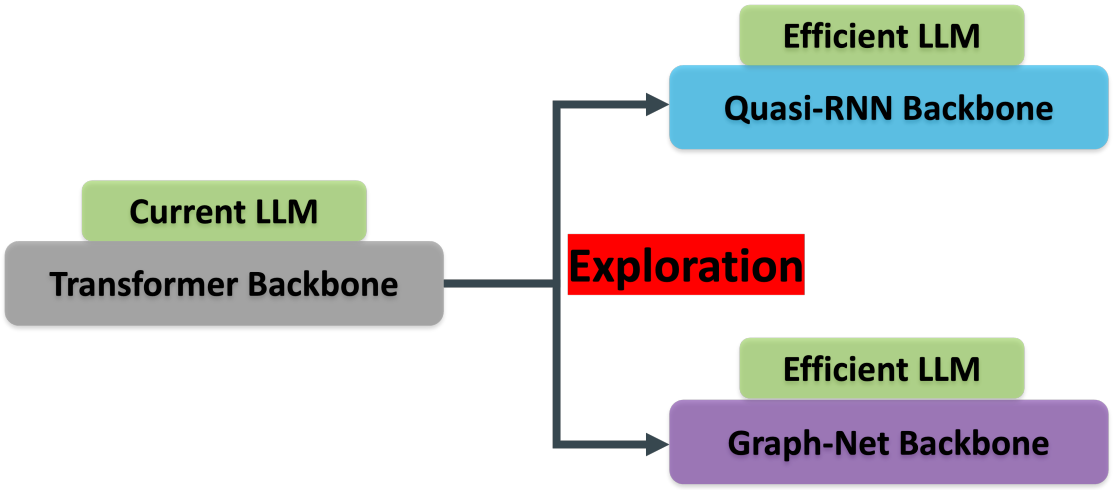


FIGURE 7.4: Exploring Sequential Recommendation-Inspired Neural Inductive Biases for Efficient Language Models.

7.2.4 Integrating Multi-Scale and Dynamic Graph Attention in Complex Recommendation Scenarios

Building on the demonstrated effectiveness of multi-scale models like QR-Rec (Chapter 3) and MSQT (Chapter 5), future work can extend these techniques to more advanced recommendation tasks, including next-basket and set recommendations. By exploring how different scales of user-item interactions influence multi-item recommendations, multi-scale frameworks can be integrated with advanced paradigms such as deep reinforcement learning or federated learning. Additionally, incorporating multi-scale designs into alternative neural architectures—like Graph Neural Networks (GNNs) or Variational Autoencoders (VAEs)—may yield deeper insights into user-item relationships across varying temporal and contextual dimensions.

Simultaneously, dynamic graph attention models, as inspired by TransGAT (Chapter 4), offer intriguing possibilities for capturing rapidly evolving user preferences in real time. Embedding temporal dynamics directly within graph structures can produce more responsive and context-aware recommendations. Further exploration of hybrid designs that combine dynamic graph attention with sequential models could unify both static and dynamic facets of user-item interactions. Testing these integrated architectures across various domains, from social media to e-commerce and content streaming, would help validate their adaptability while revealing promising new pathways for delivering highly personalized and timely recommendations.

7.2.5 Investigating Diverse Transformer Architectures in Sequential Recommendation Systems

Building upon the success of MSQT (Chapter 5) and TSNT (Chapter 6), which showcase multi-scale and nested transformer architectures, there is significant scope to delve further into transformer-based innovations for sequential recommendation systems. These methods underscore the capacity of transformers to parse intricate sequential data and uncover nuanced user-item interaction patterns. In future work, researchers could explore a wider range of transformer variants—particularly efficient transformer architectures [666]—to address the escalating demands of large-scale, real-time applications.

Adapting these advanced transformer architectures to more challenging recommendation scenarios, such as next-basket or set recommendations, could illuminate their adaptability and performance under diverse conditions. Such investigations would not only push the boundaries of modern recommendation systems but also yield broader insights into leveraging transformer-based solutions for complex sequential data. Ultimately, this line of research can catalyze the development of more accurate, context-aware, and computationally efficient recommendation systems that keep pace with ever-changing user preferences.

Bibliography

[1] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010. 2, 3, 23, 34, 49, 114, 115, 116, 129, 132, 141, 142, 143, 163, 164, 166, 192, 194, 195

[2] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 191–200. IEEE, 2016. 3, 23, 34, 49, 115, 116, 143, 163, 192

[3] Austin R Benson, Ravi Kumar, and Andrew Tomkins. Modeling user consumption sequences. In *Proceedings of the 25th International Conference on World Wide Web*, pages 519–529, 2016. 23, 49

[4] Chenwei Cai, Ruining He, and Julian McAuley. Spmc: socially-aware personalized markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497*, 2017. 23, 49

[5] Qitian Wu, Yirui Gao, Xiaofeng Gao, Paul Weng, and Guihai Chen. Dual sequential prediction models linking sequential recommendation and information dissemination. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 447–457, 2019. 2, 23, 50, 70, 132

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015. 2, 3, 23, 34, 42, 61, 105, 107, 115, 166

[7] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732. ACM, 2016. 3, 23, 34, 62, 130, 132, 142, 178, 195

[8] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503, 2017. 23, 62

[9] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 130–137. ACM, 2017. 23, 62

[10] Bjørnar Vassøy, Massimiliano Ruocco, Eliezer de Souza da Silva, and Erlend Aune. Time is of the essence: a joint hierarchical rnn and point process model for time and item predictions. In *Proceedings of the twelfth ACM international conference on Web search and data mining*, pages 591–599, 2019. 2, 23, 62

[11] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573. ACM, 2018. 2, 3, 23, 34, 42, 47, 63, 105, 108, 114, 115, 116, 117, 141, 143, 163, 164, 166, 192, 194

[12] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian McAuley. Cosrec: 2d convolutional neural networks for sequential recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2173–2176, 2019. 23, 63

[13] Trinh Xuan Tuan and Tu Minh Phuong. 3d convolutional networks for session-based recommendation with content features. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 138–146, 2017. 23, 63

[14] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 582–590. ACM, 2019. 23, 34, 63, 107, 133

[15] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenburg, and Jure Leskovec. Hierarchical temporal convolutional networks for dynamic recommender systems. In *The world wide web conference*, pages 2236–2246, 2019. 2, 23, 64

[16] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018. 2, 4, 23, 42, 69, 107, 112, 116, 143, 155, 166

[17] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. Next item recommendation with self-attention. *arXiv preprint arXiv:1808.06414*, 2018. 23, 69

[18] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326, 2019. 23, 34, 69

[19] Mingi Ji, Weonyoung Joo, Kyungwoo Song, Yoon-Yeong Kim, and Il-Chul Moon. Sequential recommendation with relation-aware kernelized self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4304–4311, 2020. 23, 69

[20] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330, 2020. 2, 23, 69, 155, 179

[21] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 346–353, 2019. 2, 23, 34, 43, 66, 72, 108, 133

[22] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 579–588, 2019. 23, 72, 133

[23] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5045–5052, 2020. 23, 73, 133

[24] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Tagnn: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1921–1924, 2020. 23, 72, 133

[25] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 169–178, 2020. 2, 23, 66, 73, 134

[26] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for next basket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 403–412, 2015. 3, 23, 34, 59, 130, 132, 142, 178

[27] Shengxian Wan, Yanyan Lan, Pengfei Wang, Jiafeng Guo, Jun Xu, and Xueqi Cheng. Next basket recommendation with neural networks. In *RecSys Posters*, 2015. 3, 23, 34, 59, 130, 132, 142

[28] Haoji Hu and Xiangnan He. Sets2sets: Learning from sequential sets with neural networks. In *Proceedings of the 25th ACM SIGKDD International*

Conference on Knowledge Discovery & Data Mining, pages 1491–1499, 2019. 3, 23, 34, 66, 176, 178, 181, 196

[29] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. An attribute-aware neural attentive model for next basket recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1201–1204, 2018. 4, 23, 34, 65, 130, 132, 142

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 4, 18, 64, 154, 166

[31] Leilei Sun, Yansong Bai, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. Dual sequential network for temporal sets prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1439–1448, 2020. 4, 23, 34, 69, 178, 181, 196

[32] Paul E Utgoff. *Machine learning of inductive bias*, volume 15. Springer Science & Business Media, 2012. 13, 14

[33] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019. 14, 28

[34] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018. 14, 28

[35] Mingming Xu, Fangai Liu, and Weizhi Xu. A survey on sequential recommendation. In *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, pages 106–111. IEEE, 2019. 28

[36] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*, 2019. 28

[37] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–42, 2020. 28, 29

[38] Qazi Mudassar Ilyas, Abid Mehmood, Ashfaq Ahmad, and Muneer Ahmad. A systematic study on a customer's next-items recommendation techniques. *Sustainability*, 14(12):7175, 2022. 14, 28

[39] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000. 14

[40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 15

[41] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986. 15

[42] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 15

[43] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 15

[44] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016. 15

[45] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017. 15

[46] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985. 16

[47] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961. 16

[48] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 16

[49] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 16

[50] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 17

[51] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998. 17, 207

[52] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 17, 59

[53] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008. 18, 207

[54] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 18, 207

[55] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 18

[56] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. Attention-based transactional context embedding for next-item recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 23, 43, 65

[57] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 23, 69

[58] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4806–4813, 2019. 23, 66

[59] Lu Yu, Chuxu Zhang, Shangsong Liang, and Xiangliang Zhang. Multi-order attentive ranking model for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5709–5716, 2019. 23, 66

[60] Kyungwoo Song, Mingi Ji, Sungrae Park, and Il-Chul Moon. Hierarchical context enabled recurrent neural network for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4983–4991, 2019. 23

[61] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428. ACM, 2017. 23, 65

[62] Shuzi Niu and Rongzhi Zhang. Collaborative sequence prediction for sequential recommender. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2239–2242, 2017. 23

[63] Yuqi Li, Weizheng Chen, and Hongfei Yan. Learning graph-based embedding for time-aware product recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2163–2166, 2017. 23, 43, 56

[64] Chen Wu and Ming Yan. Session-aware information embedding for e-commerce product recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management*, pages 2379–2382, 2017. 23, 60

[65] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 843–852, 2018. 23, 62

[66] Zhitao Wang, Chengyao Chen, Ke Zhang, Yu Lei, and Wenjie Li. Variational recurrent model for session-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1839–1842, 2018. 23

[67] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1133–1142, 2018. 23, 34, 132, 182

[68] Ke Sun, Tieyun Qian, Hongzhi Yin, Tong Chen, Yiqi Chen, and Ling Chen. What can history tell us? identifying relevant sessions for next-item recommendation. In *International Conference on Information and Knowledge Management, Proceedings*, pages 1593–1602. Association for Computing Machinery, 2019. 23

[69] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019. 23, 34, 71, 107, 154

[70] Kyeongpil Kang, Junwoo Park, Wooyoung Kim, Hojung Choe, and Jaegul Choo. Recommender system using sequential and global preference via attention mechanism and topic modeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1543–1552, 2019. 23, 69

[71] Bo Song, Yi Cao, Weifeng Zhang, and Congfu Xu. Session-based recommendation with hierarchical memory networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2181–2184, 2019. 23, 64, 72, 132, 133

[72] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. A dynamic co-attention network for session-based recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1461–1470, 2019. 23, 34, 62, 67, 108

[73] Md Mehrab Tanjim. Dynamicrec: a dynamic convolutional network for next item recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM-2020)*, 2020. 23

[74] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten De Rijke. Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1195–1204, 2020. 23

[75] Wendi Ji, Keqiang Wang, Xiaoling Wang, Tingwei Chen, and Alexandra Cristea. Sequential recommender via time-aware attentive memory network. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 565–574, 2020. 23

[76] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020. 23, 81

[77] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 175–184, 2020. 23

[78] Tong Chen, Hongzhi Yin, Hongxu Chen, Rui Yan, Quoc Viet Hung Nguyen, and Xue Li. Air: Attentional intention-aware recommender systems. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 304–315. IEEE, 2019. 23

[79] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. Context-aware sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1053–1058. IEEE, 2016. 23

[80] Tianwen Chen and Raymond Chi-Wing Wong. Session-based recommendation with local invariance. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 994–999. IEEE, 2019. 23, 69

[81] Seongwon Jang, Hoyeop Lee, Hyunsouk Cho, and Sehee Chung. Cities: Contextual inference of tail-item embeddings for sequential recommendation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 202–211. IEEE, 2020. 23

[82] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. Dgtn: Dual-channel graph transition network for session-based recommendation. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 236–242. IEEE, 2020. 23

[83] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3602–3608, 2017. 23, 62, 155

[84] Duc Trong Le, Hady W Lauw, and Yuan Fang. Basket-sensitive personalized item recommendation. IJCAI, 2017. 23, 34, 51, 132

[85] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. Diversifying personalized recommendation with user-session context. In *IJCAI*, pages 1858–1864, 2017. 23

[86] Disheng Dong, Xiaolin Zheng, Ruixun Zhang, and Yan Wang. Recurrent collaborative filtering for unifying general and sequential recommender. In *IJCAI*, pages 3350–3356, 2018. 23

[87] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*, 2018. 23, 67

[88] Ruining He, Wang-Cheng Kang, and Julian J McAuley. Translation-based recommendation: A scalable method for modeling sequential behavior. In *IJCAI*, pages 5264–5268, 2018. 23, 54

[89] Duc Trong Le, Hady W Lauw, and Yuan Fang. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. IJCAI, 2018. 23, 34, 86, 132

[90] Guibing Guo, Shichang Ouyang, Xiaodong He, Fajie Yuan, and Xiaohua Liu. Dynamic item block and prediction enhancing block for sequential recommendation. In *IJCAI*, pages 1373–1379, 2019. 23

[91] Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. Sequential and diverse recommendation with long tail. In *IJCAI*, volume 19, pages 2740–2746, 2019. 23

[92] Chenliang Li, Xichuan Niu, Xiangyang Luo, Zhenzhong Chen, and Cong Quan. A review-driven neural model for sequential recommendation. *arXiv preprint arXiv:1907.00590*, 2019. 23, 67

[93] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Longbing Cao. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *IJCAI*, pages 3771–3777, 2019. 23, 34

[94] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, pages 3940–3946, 2019. 23, 34, 69

[95] Jing Song, Hong Shen, Zijing Ou, Junyi Zhang, Teng Xiao, and Shangsong Liang. Islf: Interest shift and latent factors combination model for session-based recommendation. In *IJCAI*, pages 5765–5771, 2019. 23, 34, 66, 79

[96] Duc-Trong Le, Hady W Lauw, and Yuan Fang. Correlation-sensitive next-basket recommendation. 2019. 23, 34, 86, 132

[97] Anjing Luo, Pengpeng Zhao, Yanchi Liu, Fuzhen Zhuang, Deqing Wang, Jiajie Xu, Junhua Fang, and Victor S Sheng. Collaborative self-attention network for session-based recommendation. In *IJCAI*, pages 2591–2597, 2020. 23, 69

[98] Fei Mi and Boi Faltings. Memory augmented neural model for incremental session-based recommendation. *arXiv preprint arXiv:2005.01573*, 2020. 23, 68

[99] Ming Li, Benjamin M Dias, Ian Jarman, Wael El-Deredy, and Paulo JG Lisboa. Grocery shopping recommendations based on basket-sensitive random walk. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1215–1224, 2009. 23, 34, 57, 132

[100] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 305–314, 2017. 23, 79

[101] Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R Rabiee. Recurrent poisson factorization for temporal recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 847–855, 2017. 23

[102] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1831–1839. ACM, 2018. 23

[103] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1734–1743. ACM, 2018. 23, 62

[104] Chen Ma, Peng Kang, and Xue Liu. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 825–833, 2019. 23, 34, 86, 108

[105] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1569–1577, 2019. 23, 66

[106] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 483–491, 2020. 23, 34, 81, 86, 153

[107] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2942–2951, 2020. 23, 86

[108] Jianwen Yin, Chenghao Liu, Weiqing Wang, Jianling Sun, and Steven CH Hoi. Learning transferrable parameters for long-tailed sequential user behavior modeling. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 359–367, 2020. 23

[109] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2009–2019, 2020. 23

[110] 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. 23

[111] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 17–22. ACM, 2016. 23, 61

[112] Szu-Yu Chou, Yi-Hsuan Yang, Jyh-Shing Roger Jang, and Yu-Ching Lin. Addressing cold start for next-song recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 115–118, 2016. 23, 51

[113] Bartłomiej Twardowski. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 273–276, 2016. 23

[114] Elena Smirnova and Flavian Vasile. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 2–9. ACM, 2017. 23

[115] Sotirios P Chatzis, Panayiotis Christodoulou, and Andreas S Andreou. Recurrent latent variable networks for session-based recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pages 38–45, 2017. 23

[116] Pablo Loyola, Chen Liu, and Yu Hirate. Modeling user session and intent with an attention-based encoder-decoder architecture. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 147–151, 2017. 23, 65

[117] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 152–160, 2017. 23

[118] Rajiv Pasricha and Julian McAuley. Translation-based factorization machines for sequential recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 63–71, 2018. 23, 55

[119] Elena Smirnova. Action-conditional sequence modeling for recommendation. *arXiv preprint arXiv:1809.03291*, 2018. 23

[120] Priit Järv. Predictability limits in session-based next item recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 146–150, 2019. 23, 86

[121] Bruno L Pereira, Alberto Ueda, Gustavo Penha, Rodrygo LT Santos, and Nivio Ziviani. Online learning to rank for sequential music recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 237–245, 2019. 23, 79

[122] Heng-Shiou Sheu and Sheng Li. Context-aware graph embedding for session-based news recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 657–662, 2020. 23, 34

[123] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. Sse-pt: Sequential recommendation via personalized transformer. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 328–337, 2020. 23, 34, 154

[124] Pigi Kouki, Ilias Fountalis, Nikolaos Vasiloglou, Xiquan Cui, Edo Liberty, and Khalifeh Al Jadda. From the lab to production: a case study of session-based recommendations in the home-improvement domain. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 140–149, 2020. 23, 34

[125] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 53–62, 2020. 23, 34

[126] Jing Lin, Weike Pan, and Zhong Ming. Fissa: fusing item similarity models with self-attention networks for sequential recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 130–139, 2020. 23, 34

[127] Sung Min Cho, Eunhyeok Park, and Sungjoo Yoo. Meantime: Mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 515–520, 2020. 23, 34, 154, 155, 182

[128] Andres Ferraro, Dietmar Jannach, and Xavier Serra. Exploring longitudinal effects of session-based recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 474–479, 2020. 23

[129] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 505–514. ACM, 2018. 23, 68, 75

[130] Xiang Lin, Shuzi Niu, Yiqiao Wang, and Yucheng Li. K-plet recurrent neural networks for sequential recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1057–1060, 2018. 23, 62

[131] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 675–684, 2019. 23, 64, 70

[132] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 345–354, 2019. 23, 68

[133] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. π-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 685–694, 2019. 23, 62

[134] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Sequence and time aware neighborhood for session-based recommendations: Stan. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1069–1072, 2019. 23, 53

[135] Lei Zheng, Ziwei Fan, Chun-Ta Lu, Jiawei Zhang, and Philip S Yu. Gated spectral units: Modeling co-evolving patterns for sequential recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1077–1080, 2019. 23, 86

[136] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 565–574, 2019. 23, 68

[137] Wen Wang, Wei Zhang, Jun Rao, Zhijie Qiu, Bo Zhang, Leyu Lin, and Hongyuan Zha. Group-aware long-and short-term graph representation learning for sequential group recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1449–1458, 2020. 23, 69, 73, 134

[138] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1110, 2020. 23, 69, 73, 133

[139] Wenjing Meng, Deqing Yang, and Yanghua Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval*, pages 1091–1100, 2020. 23, 73, 134

[140] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 109–118, 2020. 23, 75, 156

[141] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, Shaozhang Niu, and Jimmy Huang. Kerl: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 209–218, 2020. 23, 75, 79

[142] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 89–98, 2020. 23, 69, 76

[143] Yang Sun, Fajie Yuan, Min Yang, Guoao Wei, Zhou Zhao, and Duo Liu. A generic network compression framework for sequential recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1299–1308, 2020. 23, 43, 64

[144] Shihao Li, Dekun Yang, and Bufeng Zhang. Mrif: Multi-resolution interest fusion for recommendation. In *Proceedings of the 43rd International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 1765–1768, 2020. 23, 69

[145] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. Rethinking item importance in session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1837–1840, 2020. 23

[146] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. An intent-guided collaborative machine for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1833–1836, 2020. 23, 68

[147] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. Time matters: Sequential recommendation with complex temporal information. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1459–1468, 2020. 23, 70, 155, 182

[148] Cheng Guo, Mengfei Zhang, Jinyun Fang, Jiaqi Jin, and Mao Pan. Session-based recommendation with hierarchical leaping networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1705–1708, 2020. 23, 62

[149] Jarana Manotumruksa and Emine Yilmaz. Sequential-based adversarial optimisation for personalised top-n item recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2045–2048, 2020. 23, 76

[150] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 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*, pages 669–678, 2020. 23, 69, 73, 133

[151] Lin Zheng, Naicheng Guo, Weihao Chen, Jin Yu, and Dazhi Jiang. Sentiment-guided sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1957–1960, 2020. 23, 69

[152] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1469–1478, 2020. 23, 63

[153] Madiraju Srilakshmi, Gourab Chowdhury, and Sudeshna Sarkar. Improved session based recommendation using graph-based item embedding. In *SIGIR 2020 workshop on eCommerce*, 2020. 23, 75

[154] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1071–1080, 2020. 23, 34, 53, 132

[155] How Jing and Alexander J Smola. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 515–524. ACM, 2017. 23, 62

[156] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 108–116. ACM, 2018. 23, 68

[157] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 573–581, 2019. 23

[158] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. Sequential variational autoencoders for collaborative filtering. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 600–608, 2019. 23, 79

[159] Yun He, Yin Zhang, Weiwen Liu, and James Caverlee. Consistency-aware recommendation for user-generated item list continuation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 250–258, 2020. 23, 66

[160] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. Lara: Attribute-to-feature adversarial learning for new-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 582–590, 2020. 23, 76

[161] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 645–653, 2020. 23, 66

[162] Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. Sequential modeling of hierarchical user intention and preference for next-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 807–815, 2020. 23, 68

[163] Jiarui Qin, Kan Ren, Yuchen Fang, Weinan Zhang, and Yong Yu. Sequential recommendation with dual side neighbor-based collaborative relation modeling. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 465–473, 2020. 23, 67

[164] Lakshmanan Rakkappan and Vaibhav Rajan. Context-aware sequential recommendations with stacked recurrent neural networks. In *The World Wide Web Conference*, pages 3172–3178, 2019. 23

[165] Teng Xiao, Shangsong Liang, and Zaiqiao Meng. Hierarchical neural variational model for personalized sequential recommendation. In *The World Wide Web Conference*, pages 3377–3383, 2019. 23

[166] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. Recurrent convolutional neural network for sequential recommendation. In *The world wide web conference*, pages 3398–3404, 2019. 23

[167] Fan Zhou, Zijing Wen, Kunpeng Zhang, Goce Trajcevski, and Ting Zhong. Variational session-based recommendation using normalizing flows. In *The World Wide Web Conference*, pages 3476–3475, 2019. 23, 79

[168] Xueliang Guo, Chongyang Shi, and Chuanming Liu. Intention modeling from ordered and unordered facets for sequential recommendation. In *Proceedings of The Web Conference 2020*, pages 1127–1137, 2020. 23, 66

[169] Jibang Wu, Renqin Cai, and Hongning Wang. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*, pages 2199–2209, 2020. 23, 69, 182

[170] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of The Web Conference 2020*, pages 303–313, 2020. 23, 34, 64, 153

[171] Yuanxing Liu, Zhaochun Ren, Wei-Nan Zhang, Wanxiang Che, Ting Liu, and Dawei Yin. Keywords generation improves e-commerce session-based recommendation. In *Proceedings of The Web Conference 2020*, pages 1604–1614, 2020. 23, 69

[172] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*, pages 2528–2534, 2020. 23, 69

[173] Yin Zhang, Yun He, Jianling Wang, and James Caverlee. Adaptive hierarchical translation-based sequential recommendation. In *Proceedings of The Web Conference 2020*, pages 2984–2990, 2020. 23, 55

[174] Hengtong Zhang, Yaliang Li, Bolin Ding, and Jing Gao. Practical data poisoning attack against next-item recommendation. In *Proceedings of The Web Conference 2020*, pages 2458–2464, 2020. 23, 79

[175] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. Mv-rnn: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):317–331, 2018. 23

[176] Seyed Abbas Hosseini, Ali Khodadadi, Keivan Alizadeh, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R Rabiee. Recurrent poisson factorization for temporal recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):121–134, 2018. 23, 51

[177] Hui Li, Ye Liu, Nikos Mamoulis, and David S Rosenblum. Translation-based sequential recommendation for complex users on sparse data. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1639–1651, 2019. 23

[178] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3946–3957, 2020. 23

[179] Xiaolin Zheng, Menghan Wang, Renjun Xu, Jianmeng Li, and Yan Wang. Modeling dynamic missingness of implicit feedback for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):405–418, 2020. 23

[180] Zijie Zeng, Jing Lin, Lin Li, Weike Pan, and Zhong Ming. Next-item recommendation via collaborative filtering with bidirectional item similarity. *ACM Transactions on Information Systems (TOIS)*, 38(1):1–22, 2019. 23

[181] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–33, 2020. 23

[182] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–23, 2020. 23

[183] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4503–4511, 2021. 24

[184] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4123–4130, 2021. 24

[185] Lei Chen, Fajie Yuan, Jiaxi Yang, Xiang Ao, Chengming Li, and Min Yang. A user-adaptive layer selection framework for very deep sequential recommender models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3984–3991, 2021. 24

[186] Jiahao Yuan, Zihan Song, Mingyou Sun, Xiaoling Wang, and Wayne Xin Zhao. Dual sparse attention network for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4635–4643, 2021. 24

[187] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. Noninvasive self-attention for side information fusion in sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4249–4256, 2021. 24

[188] Qiaoyu Tan, Jianwei Zhang, Ninghao Liu, Xiao Huang, Hongxia Yang, Jingren Zhou, and Xia Hu. Dynamic memory based attention network for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4384–4392, 2021. 24

[189] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. Cold-start sequential recommendation via meta learner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4706–4713, 2021. 24

[190] Jongwon Jeong, Jeong Choi, Hyunsouk Cho, and Sehee Chung. Fpadametric: False-positive-aware adaptive metric learning for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4039–4047, 2022. 24

[191] Aritra Ghosh, Saayan Mitra, and Andrew Lan. Dips: Differentiable policy for sketching in recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6703–6712, 2022. 24

[192] Michael Potter, Hamlin Liu, Yash Lala, Christian Loanzon, and Yizhou Sun. Gru4recbe: a hybrid session-based movie recommendation system (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 13029–13030, 2022. 24

[193] Jiayu Song, Jiajie Xu, Rui Zhou, Lu Chen, Jianxin Li, and Chengfei Liu. Cbml: A cluster-based meta-learning model for session-based recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1713–1722, 2021. 24

[194] Qitian Wu, Chenxiao Yang, Shuodian Yu, Xiaofeng Gao, and Guihai Chen. Seq2bubbles: Region-based embedding learning for user behaviors in sequential recommenders. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2160–2169, 2021. 24

[195] Chunyang Wang, Yanmin Zhu, Haobing Liu, Wenze Ma, Tianzi Zang, and Jiadi Yu. Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1889–1898, 2021. 24

[196] Yunzhe Li, Yue Ding, Bo Chen, Xin Xin, Yule Wang, Yuxiang Shi, Ruiming Tang, and Dong Wang. Extracting attentive social temporal excitation for sequential recommendation. *arXiv preprint arXiv:2109.13539*, 2021. 24

[197] Yicong Li, Hongxu Chen, Xiangguo Sun, Zhenchao Sun, Lin Li, Lizhen Cui, Philip S Yu, and Guandong Xu. Hyperbolic hypergraphs for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 988–997, 2021. 24

[198] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. Lightweight self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 967–977, 2021. 24

[199] Zhankui He, Handong Zhao, Zhe Lin, Zhaowen Wang, Ajinkya Kale, and Julian McAuley. Locker: Locally constrained self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3088–3092, 2021. 24

[200] Zeyuan Chen, Wei Zhang, Junchi Yan, Gang Wang, and Jianyong Wang. Learning dual dynamic representations on time-sliced user-item interaction graphs for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 231–240, 2021. 24

[201] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. Self-supervised graph co-training for session-based recommendation. In *Proceedings of the 30th ACM International conference on information & knowledge management*, pages 2180–2190, 2021. 24

[202] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 433–442, 2021. 24, 154, 155

[203] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S Yu. Modeling sequences as distributions with uncertainty for sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3019–3023, 2021. 24, 154

[204] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Causer: Causal session-based recommendations for handling popularity bias. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3048–3052, 2021. 24

[205] Hengyu Zhang, Enming Yuan, Wei Guo, Zhicheng He, Jiarui Qin, Huifeng Guo, Bo Chen, Xiu Li, and Ruiming Tang. Disentangling past-future modeling in sequential recommendation via dual networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2549–2558, 2022. 24

[206] Dongmin Hyun, Chanyoung Park, Junsu Cho, and Hwanjo Yu. Beyond learning from next item: Sequential recommendation via personalized interest sustainability. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 812–821, 2022. 24

[207] Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. Explanation guided contrastive learning for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2017–2027, 2022. 24

[208] Ziyang Wang, Huoyu Liu, Wei Wei, Yue Hu, Xian-Ling Mao, Shaojian He, Rui Fang, and Dangyang Chen. Multi-level contrastive learning framework for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2098–2107, 2022. 24

[209] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2056–2066, 2022. 24

[210] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3953–3957, 2022. 24

[211] Hanwen Du, Hui Shi, Pengpeng Zhao, Deqing Wang, Victor S Sheng, Yanchi Liu, Guanfeng Liu, and Lei Zhao. Contrastive learning with bidirectional transformers for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 396–405, 2022. 24, 154

[212] Lihua Chen, Ning Yang, and Philip S Yu. Time lag aware sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 212–221, 2022. 24, 155

[213] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 138–147, 2022. 24

[214] Jiayan Guo, Peiyan Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. Evolutionary preference learning via graph nested gru ode for session-based recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 624–634, 2022. 24

[215] Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. Spatiotemporal-aware session-based recommendation with graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1209–1218, 2022. 24

[216] Jinwei Luo, Mingkai He, Xiaolin Lin, Weike Pan, and Zhong Ming. Dual-task learning for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 1379–1388, 2022. 24

[217] Changxin Tian, Zihan Lin, Shuqing Bian, Jinpeng Wang, and Wayne Xin Zhao. Temporal contrastive pre-training for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1925–1934, 2022. 24

[218] Chen Xu, Jun Xu, Xu Chen, Zhenghua Dong, and Ji-Rong Wen. Dually enhanced propensity score estimation in sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2260–2269, 2022. 24

[219] Chi Zhang, Yantong Du, Xiangyu Zhao, Qilong Han, Rui Chen, and Li Li. Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2508–2518, 2022. 24

[220] Kesen Zhao, Xiangyu Zhao, Zijian Zhang, and Muyang Li. Mae4rec: Storage-saving transformer for sequential recommendations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2681–2690, 2022. 24, 154

[221] Shuqing Bian, Wayne Xin Zhao, Jinpeng Wang, and Ji-Rong Wen. A relevant and diverse retrieval-enhanced data augmentation framework for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2923–2932, 2022. 24

[222] Xingyu Pan, Yushuo Chen, Changxin Tian, Zihan Lin, Jinpeng Wang, He Hu, and Wayne Xin Zhao. Multimodal meta-learning for cold-start sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 3421–3430, 2022. 24

[223] Zhankui He, Handong Zhao, Zhaowen Wang, Zhe Lin, Ajinkya Kale, and Julian Mcauley. Query-aware sequential recommendation. In *Proceedings of*

the 31st ACM International Conference on Information & Knowledge Management, pages 4019–4023, 2022. 24

[224] Sejoon Oh, Ankur Bhardwaj, Jongseok Han, Sungchul Kim, Ryan A Rossi, and Srijan Kumar. Implicit session contexts for next-item recommendations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4364–4368, 2022. 24

[225] Hao Qian, Qintong Wu, MingHao Li, Zhengwei Wu, Zhiqiang Zhang, Jun Zhou, Lihong Gu, and Jinjie Gu. Fwseqblock: A field-wise approach for modeling behavior representation in sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4404–4408, 2022. 24

[226] Kazutoshi Umemoto. Ml-1m++: Movielens-compatible additional preferences for more robust offline evaluation of sequential recommenders. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4540–4544, 2022. 24

[227] Xianghong Xu, Kai Ouyang, Liuyin Wang, Jiaxin Zou, Yanxiong Lu, Hai-Tao Zheng, and Hong-Gee Kim. Modeling latent autocorrelation for session-based recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4605–4609, 2022. 24

[228] Jing Zhao, Pengpeng Zhao, Lei Zhao, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. Variational self-attention network for sequential recommendation. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1559–1570. IEEE, 2021. 24

[229] Tao Xie, Yangjun Xu, Liang Chen, Yang Liu, and Zibin Zheng. Sequential recommendation on dynamic heterogeneous information network. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2105–2110. IEEE, 2021. 24

[230] Liuyin Wang, Xianghong Xu, Kai Ouyang, Huanzhong Duan, Yanxiong Lu, and Hai-Tao Zheng. Self-supervised dual-channel attentive network for session-based social recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2034–2045. IEEE, 2022. 24

[231] Jiahao Yuan, Wendi Ji, Dell Zhang, Jinwei Pan, and Xiaoling Wang. Micro-behavior encoding for session-based recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2886–2899. IEEE, 2022. 24

[232] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022. 24, 81

[233] Zizhuo Zhang and Bang Wang. Graph neighborhood routing and random walk for session-based recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1517–1522. IEEE, 2021. 24

[234] Ruihong Qiu, Zi Huang, and Hongzhi Yin. Memory augmented multi-instance contrastive predictive coding for sequential recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 519–528. IEEE, 2021. 24

[235] Zhiding Liu, Mingyue Cheng, Zhi Li, Qi Liu, and Enhong Chen. One person, one model—learning compound router for sequential recommendation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 289–298. IEEE, 2022. 24

[236] Ruihong Qiu, Zi Huang, and Hongzhi Yin. Beyond double ascent via recurrent neural tangent kernel in sequential recommendation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 428–437. IEEE, 2022. 24

[237] Chao Chen, Haoyu Geng, Nianzu Yang, Junchi Yan, Daiyue Xue, Jianping Yu, and Xiaokang Yang. Learning self-modulating attention in continuous time space with applications to sequential recommendation. In *International Conference on Machine Learning*, pages 1606–1616. PMLR, 2021. 24

[238] Yatong Sun, Bin Wang, Zhu Sun, and Xiaochun Yang. Does every data instance matter? enhancing sequential recommendation by eliminating unreliable data. In *IJCAI*, pages 1579–1585, 2021. 24

[239] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. Da-gcn: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. *arXiv preprint arXiv:2105.03300*, 2021. 24

[240] Xiaohai Tong, Pengfei Wang, Chenliang Li, Long Xia, and Shaozhang Niu. Pattern-enhanced contrastive policy learning network for sequential recommendation. In *IJCAI*, pages 1593–1599, 2021. 24

[241] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, Zhuoye Ding, Zhen He, and Bo Long. Improving sequential recommendation consistency with self-supervised imitation. *arXiv preprint arXiv:2106.14031*, 2021. 24

[242] Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. Exploring periodicity and interactivity in multi-interest framework for sequential recommendation. *arXiv preprint arXiv:2106.04415*, 2021. 24

[243] Pengyu Zhao, Tianxiao Shui, Yuanxing Zhang, Kecheng Xiao, and Kaigui Bian. Adversarial oracular seq2seq learning for sequential recommendation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1905–1911, 2021. 24, 34, 76, 153

[244] Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. Enhancing sequential recommendation with graph contrastive learning. *arXiv preprint arXiv:2205.14837*, 2022. 24

[245] Muyang Li, Xiangyu Zhao, Chuan Lyu, Minghao Zhao, Runze Wu, and Ruocheng Guo. Mlp4rec: A pure mlp architecture for sequential recommendations. *arXiv preprint arXiv:2204.11510*, 2022. 24

[246] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3161–3171, 2021. 24

[247] Ehsan Gholami, Mohammad Motamedi, and Ashwin Aravindakshan. Parsrec: Explainable personalized attention-fused recurrent sequential recommendation using session partial actions. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 454–464, 2022. 24

[248] Khalil Damak, Sami Khenissi, and Olfa Nasraoui. Debiasing the cloze task in sequential recommendation with bidirectional transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 273–282, 2022. 24

[249] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2263–2274, 2022. 24, 154

[250] Zihan Wang, Na Huang, Fei Sun, Pengjie Ren, Zhumin Chen, Hengliang Luo, Maarten de Rijke, and Zhaochun Ren. Debiasing learning for membership inference attacks against recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1959–1968, 2022. 24

[251] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593, 2022. 24

[252] Chenxiao Yang, Qitian Wu, Qingsong Wen, Zhiqiang Zhou, Liang Sun, and Junchi Yan. Towards out-of-distribution sequential event prediction: A causal treatment. *Advances in neural information processing systems*, 35: 22656–22670, 2022. 24

[253] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 143–153, 2021. 24, 154

[254] Zhenrui Yue, Zhankui He, Huimin Zeng, and Julian McAuley. Black-box attacks on sequential recommenders via data-free model extraction. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 44–54, 2021. 24

[255] Vojtěch Vančura. Neural basket embedding for sequential recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 878–883, 2021. 24, 34, 132, 182

[256] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 505–514, 2021. 24

[257] Wenzhuo Song, Shoujin Wang, Yan Wang, and Shengsheng Wang. Next-item recommendations in short sessions. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 282–291, 2021. 24

[258] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 92–101, 2022. 24, 154

[259] Ori Katz, Oren Barkan, Noam Koenigstein, and Nir Zabari. Learning to ride a buy-cycle: A hyper-convolutional model for next basket repurchase recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 316–326, 2022. 24, 34, 132

[260] Weixin Chen, Mingkai He, Yongxin Ni, Weike Pan, Li Chen, and Zhong Ming. Global and personalized graphs for heterogeneous sequential recommendation by learning behavior transitions and user intentions. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 268–277, 2022. 24

[261] Wei Cai, Weike Pan, Jingwen Mao, Zhechao Yu, and Congfu Xu. Aspect redistribution for learning better item embeddings in sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 49–58, 2022. 24

[262] Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. Defending substitution-based profile pollution attacks on sequential recommenders. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 59–70, 2022. 24

[263] Aleksandr Petrov and Craig Macdonald. A systematic review and replicability study of bert4rec for sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 436–447, 2022. 24

[264] Aleksandr Petrov and Craig Macdonald. Effective and efficient training for sequential recommendation using recency sampling. In *Proceedings of the*

16th ACM Conference on Recommender Systems, pages 81–91, 2022. 24, 34, 153

[265] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. Context and attribute-aware sequential recommendation via cross-attention. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 71–80, 2022. 24

[266] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022. 24, 83

[267] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. Temporal augmented graph neural networks for session-based recommendations. In *Proceedings of the 44th International ACM SIGIR conference on research and development in information retrieval*, pages 1798–1802, 2021. 24, 182

[268] Taegwan Kang, Hwanhee Lee, Byeongjin Choe, and Kyomin Jung. Entangled bidirectional encoder to autoregressive decoder for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1657–1661, 2021. 24, 34, 153

[269] Xu Yuan, Dongsheng Duan, Lingling Tong, Lei Shi, and Cheng Zhang. Icai-sr: Item categorical attribute integrated sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1687–1691, 2021. 24

[270] Junsu Cho, SeongKu Kang, Dongmin Hyun, and Hwanjo Yu. Unsupervised proxy selection for session-based recommender systems. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 327–336, 2021. 24

[271] Chen Chen, Jie Guo, and Bin Song. Dual attention transfer in session-based recommendation with multi-dimensional integration. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 869–878, 2021. 24

[272] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 367–377, 2021. 24

[273] Jianling Wang, Kaize Ding, and James Caverlee. Sequential recommendation for cold-start users with meta transitional learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1783–1787, 2021. 24

[274] Zeyu Cui, Yinjiang Cai, Shu Wu, Xibo Ma, and Liang Wang. Motif-aware sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1738–1742, 2021. 24

[275] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 347–356, 2021. 24

[276] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. Category-aware collaborative sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 388–397, 2021. 24

[277] Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1733–1737, 2021. 24

[278] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 378–387, 2021. 24

[279] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, pages 1608–1612, 2021. 24, 154

[280] Jiachun Wang, Fajie Yuan, Jian Chen, Qingyao Wu, Min Yang, Yang Sun, and Guoxiao Zhang. Stackrec: Efficient training of very deep sequential recommender models by iterative stacking. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 357–366, 2021. 24

[281] Yuqi Qin, Pengfei Wang, and Chenliang Li. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 859–868, 2021. 24, 34, 133

[282] Kaiyuan Li, Pengfei Wang, and Chenliang Li. Multi-agent rl-based information selection model for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1622–1631, 2022. 24

[283] Jie Zou, Evangelos Kanoulas, Pengjie Ren, Zhaochun Ren, Aixin Sun, and Cheng Long. Improving conversational recommender systems via transformer-based sequential modelling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2319–2324, 2022. 24

[284] Lei Chen, Jingtao Ding, Min Yang, Chengming Li, Chonggang Song, and Lingling Yi. Item-provider co-learning for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1817–1822, 2022. 24

[285] Guanyu Lin, Chen Gao, Yinfeng Li, Yu Zheng, Zhiheng Li, Depeng Jin, and Yong Li. Dual contrastive network for sequential recommendation with user and item-centric perspectives. *arXiv preprint arXiv:2209.08446*, 2022. 24

[286] Yuehua Zhu, Bo Huang, Shaohua Jiang, Muli Yang, Yanhua Yang, and Wenliang Zhong. Progressive self-attention network with unsymmetrical positional encoding for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2029–2033, 2022. 24

[287] Mozhdeh Ariannezhad, Sami Jullien, Ming Li, Min Fang, Sebastian Schelter, and Maarten de Rijke. Recanet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1240–1250, 2022. 24, 34, 133

[288] Qihang Zhao. Resetbert4rec: A pre-training model integrating time and user historical behavior for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1812–1816, 2022. 24, 155

[289] Xinyan Fan, Jianxun Lian, Wayne Xin Zhao, Zheng Liu, Chaozhuo Li, and Xing Xie. Ada-ranker: A data distribution adaptive ranking paradigm for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1599–1610, 2022. 24

[290] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Ao Xiang, Xu Zhang, Leyu Lin, and Qing He. Selective fairness in recommendation via prompts. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2657–2662, 2022. 24

[291] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1632–1641, 2022. 24

[292] Yuli Liu, Christian Walder, and Lexing Xie. Determinantal point process likelihoods for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1653–1663, 2022. 24

[293] Yueqi Xie, Peilin Zhou, and Sunghun Kim. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1611–1621, 2022. 24

[294] Jinseok Jamie Seol, Youngrok Ko, and Sang-goo Lee. Exploiting session information in bert-based session-aware sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2639–2644, 2022. 24

[295] Yongjun Chen, Jia Li, and Caiming Xiong. Elecrec: Training sequential recommenders as discriminators. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2550–2554, 2022. 24

[296] Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian McAuley. Coarse-to-fine sparse sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2082–2086, 2022. 24

[297] Chuhan Wu, Fangzhao Wu, Tao Qi, Chenliang Li, and Yongfeng Huang. Is news recommendation a sequential recommendation task? In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2382–2386, 2022. 24

[298] Enming Yuan, Wei Guo, Zhicheng He, Huifeng Guo, Chengkai Liu, and Ruiming Tang. Multi-behavior sequential transformer recommender. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1642–1652, 2022. 24, 154

[299] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 598–606, 2021. 24

[300] Tianwen Chen and Raymond Chi-Wing Wong. An efficient and effective framework for session-based social recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 400–408, 2021. 24, 73, 134

[301] Chenyang Wang. Towards dynamic user intention in sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1121–1122, 2021. 24

[302] Minjin Choi, Jinhong Kim, Joonseok Lee, Hyunjung Shim, and Jongwuk Lee. S-walk: accurate and scalable session-based recommendation with random walks. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 150–160, 2022. 24, 57

[303] Jiayan Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. Learning multi-granularity consecutive user intent unit for session-based recommendation. In *Proceedings of the fifteenth ACM International conference on web search and data mining*, pages 343–352, 2022. 24

[304] Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, Beibei Kong, and Di Niu. Recguru: Adversarial learning of generalized user representations for cross-domain recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 571–581, 2022. 24, 76

[305] Uriel Singer, Haggai Roitman, Yotam Eshel, Alexander Nus, Ido Guy, Or Levi, Idan Hasson, and Eliyahu Kiperwasser. Sequential modeling with multiple attributes for watchlist recommendation in e-commerce. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 937–946, 2022. 24, 70

[306] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823, 2022. 24, 81

[307] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 775–783, 2022. 24, 73, 134

[308] Naime Ranjbar Kermany, Jian Yang, Jia Wu, and Luiz Pizzato. Fair-srs: a fair session-based recommendation system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1601–1604, 2022. 24

[309] Minjin Choi, Jinhong Kim, Joonseok Lee, Hyunjung Shim, and Jongwuk Lee. Session-aware linear item-item models for session-based recommendation. In *Proceedings of the Web Conference 2021*, pages 2186–2197, 2021. 24, 34

[310] Jialiang Han, Yun Ma, Qiaozhu Mei, and Xuanzhe Liu. Deeprec: On-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In *Proceedings of the Web Conference 2021*, pages 900–911, 2021. 24, 34

[311] Yongji Wu, Defu Lian, Neil Zhenqiang Gong, Lu Yin, Mingyang Yin, Jingren Zhou, and Hongxia Yang. Linear-time self attention with codeword histogram for efficient recommendation. In *Proceedings of the Web Conference 2021*, pages 1262–1273, 2021. 24, 34, 70

[312] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the Web Conference 2021*, pages 449–459, 2021. 24, 34, 79

[313] Cheng Hsu and Cheng-Te Li. Retagnn: relational temporal attentive graph neural networks for holistic sequential recommendation. In *Proceedings of the web conference 2021*, pages 2968–2979, 2021. 24, 34, 73, 134

[314] Pedro Dalla Vecchia Chaves, Bruno L Pereira, and Rodrygo LT Santos. Efficient online learning to rank for sequential music recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2442–2450, 2022. 24, 34

[315] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2388–2399, 2022. 24, 34, 60

[316] Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Disentangling long and short-term interests for recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2256–2267, 2022. 24, 34, 81

[317] Jiarui Jin, Xianyu Chen, Weinan Zhang, Junjie Huang, Ziming Feng, and Yong Yu. Learn over past, evolve for future: Search-based time-aware recommendation with sequential behavior data. In *Proceedings of the ACM Web Conference 2022*, pages 2451–2461, 2022. 24, 34

[318] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182, 2022. 24, 34, 81

[319] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM Web Conference 2022*, pages 2036–2047, 2022. 24, 34, 70, 154

[320] Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, and Xia Ning. Ham: Hybrid associations models for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4838–4853, 2021. 24, 34

[321] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5403–5414, 2021. 24, 34

[322] Tianyu Zhu, Leilei Sun, and Guoqing Chen. Graph-based embedding smoothing for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):496–508, 2021. 24

[323] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. Incorporating link prediction into multi-relational item graph modeling for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2683–2696, 2021. 24, 34

[324] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Parallel split-join networks for shared account cross-domain sequential recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4106–4123, 2021. 24, 34

[325] Lei Guo, Jinyu Zhang, Tong Chen, Xinhua Wang, and Hongzhi Yin. Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34

[326] Ansong Li, Zhiyong Cheng, Fan Liu, Zan Gao, Weili Guan, and Yuxin Peng. Disentangled graph neural networks for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34

[327] Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. Multi-behavior sequential recommendation with temporal graph transformer. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34, 154, 155, 182

[328] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34

[329] Chao Chen, Dongsheng Li, Junchi Yan, and Xiaokang Yang. Modeling dynamic user preference via dictionary learning for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5446–5458, 2022. 24, 34

[330] Shilin Qu, Fajie Yuan, Guibing Guo, Liguang Zhang, and Wei Wei. Cmnrec: Sequential recommendations with chunk-accelerated memory network. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3540–3550, 2022. 24, 34

[331] Hengtong Zhang, Yaliang Li, Bolin Ding, and Jing Gao. Loki: a practical data poisoning attack framework against next item recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5047–5059, 2022. 24, 34

[332] Zhizhuo Yin, Kai Han, Pengzi Wang, and Haibing Hu. Multi global information assisted streaming session-based recommendation system. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34

[333] Xu Chen, Zhenlei Wang, Hongteng Xu, Jingsen Zhang, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Data augmented sequential recommendation based on counterfactual thinking. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 24, 34

[334] Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, and Xia Ning. M 2: Mixed models with preferences, popularities and transitions for next-basket recommendation. *IEEE transactions on knowledge and data engineering*, 35(4): 4033–4046, 2022. 24, 34, 133

[335] Wei Wang and Longbing Cao. Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–26, 2021. 24, 34, 133

[336] Yanan Xu, Yanmin Zhu, and Jiadi Yu. Modeling multiple coexisting category-level intentions for next item recommendation. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–24, 2021. 24

[337] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten De Rijke. Multi-interest diversification for end-to-end sequential recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–30, 2021. 24

[338] Wei Zhang, Zeyuan Chen, Hongyuan Zha, and Jianyong Wang. Learning from substitutable and complementary relations for graph-based sequential product recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–28, 2021. 24

[339] Ruihong Qiu, Zi Huang, Tong Chen, and Hongzhi Yin. Exploiting positional information for session-based recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–24, 2021. 24

[340] Nengjun Zhu, Jian Cao, Xinjiang Lu, and Hui Xiong. Learning a hierarchical intent model for next-item recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–28, 2021. 24

[341] Zhiqiang Pan, Fei Cai, Wanyu Chen, and Honghui Chen. Graph co-attentive session-based recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–31, 2021. 24

[342] Zhiqiang Pan, Fei Cai, Wanyu Chen, Chonghao Chen, and Honghui Chen. Collaborative graph learning for session-based recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–26, 2022. 24

[343] Xiaowen Huang, Jitao Sang, Jian Yu, and Changsheng Xu. Learning to learn a cold-start sequential recommender. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–25, 2022. 24

[344] Junsu Cho, Dongmin Hyun, Dong won Lim, Hyeon jae Cheon, Hyoung-iel Park, and Hwanjo Yu. Dynamic multi-behavior sequence modeling for next item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4199–4207, 2023. 26, 34

[345] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4225–4232, 2023. 26, 34, 156

[346] Lei Li, Jianxun Lian, Xiao Zhou, and Xing Xie. Ada-retrieval: An adaptive multi-round retrieval paradigm for sequential recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8670–8678, 2024. 26

[347] Refaldi Putra and Toyotaro Suzumura. On the role of numerical encoding in foundation model of sequential recommendation with sequential indexing. In *Workshop on Recommendation Ecosystems: Modeling, Optimization and Incentive Design*, 2024. 26

[348] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. Plug-in diffusion model for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8886–8894, 2024. 26

[349] Haitong Luo, Xuying Meng, Suhang Wang, Hanyun Cao, Weiyao Zhang, Yequan Wang, and Yujun Zhang. Spectral-based graph neutral networks for complementary item recommendation. *arXiv preprint arXiv:2401.02130*, 2024. 26

[350] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8984–8992, 2024. 26

[351] Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Dual-view whitening on pre-trained text embeddings for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9332–9340, 2024. 26

[352] Junyang Chen, Guoxuan Zou, Pan Zhou, Wu Yirui, Zhenghan Chen, Houcheng Su, Huan Wang, and Zhiguo Gong. Sparse enhanced network: An adversarial generation method for robust augmentation in sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8283–8291, 2024. 26

[353] Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9359–9367, 2024. 26, 183

[354] Kexin Li, Chengjiang Long, Shengyu Zhang, Xudong Tang, Zhichao Zhai, Kun Kuang, and Jun Xiao. Corerec: A counterfactual correlation inference

for next set recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8661–8669, 2024. 26

[355] Qingtian Bian, Jiaxing Xu, Hui Fang, and Yiping Ke. Cpmr: Context-aware incremental sequential recommendation with pseudo-multi-task learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 120–130, 2023. 26

[356] Dui Wang, Xiangyu Hou, Xiaohui Yang, Bo Zhang, Renbing Chen, and Daiyue Xue. Multiple key-value strategy in recommendation systems incorporating large language model. *arXiv preprint arXiv:2310.16409*, 2023. 26, 83, 85

[357] Shuqing Bian, Xingyu Pan, Wayne Xin Zhao, Jinpeng Wang, Chuyuan Wang, and Ji-Rong Wen. Multi-modal mixture of experts represetation learning for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 110–119, 2023. 26

[358] Yujia Ding, Huan Li, Ke Chen, and Lidan Shou. Tpuf: Enhancing cross-domain sequential recommendation via transferring pre-trained user features. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 410–419, 2023. 26

[359] Mingjia Yin, Hao Wang, Xiang Xu, Likang Wu, Sirui Zhao, Wei Guo, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. Apgl4sr: A generic framework with adaptive and personalized global collaborative information in sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3009–3019, 2023. 26

[360] Chuang Zhao, Xinyu Li, Ming He, Hongke Zhao, and Jianping Fan. Sequential recommendation via an adaptive cross-domain knowledge decomposition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3453–3463, 2023. 26

[361] Changxin Tian, Binbin Hu, Wayne Xin Zhao, Zhiqiang Zhang, and Jun Zhou. Periodicity may be emanative: Hierarchical contrastive learning for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2442–2451, 2023. 26

[362] Chung Park, Taesan Kim, Taekyoon Choi, Junui Hong, Yelim Yu, Mincheol Cho, Kyunam Lee, Sungil Ryu, Hyungjun Yoon, Minsung Choi, et al. Cracking the code of negative transfer: A cooperative game theoretic approach for cross-domain sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2024–2033, 2023. 26

[363] Juyong Jiang, Peiyan Zhang, Yingtao Luo, Chaozhuo Li, Jae Boum Kim, Kai Zhang, Senzhang Wang, Xing Xie, and Sunghun Kim. Adamct: adaptive mixture of cnn-transformer for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 976–986, 2023. 26, 154

[364] Sijia Liu, Jiahao Liu, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. Autoseqrec: Autoencoder for efficient sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1493–1502, 2023. 26

[365] Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. Text matching improves sequential recommendation by reducing popularity biases. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1534–1544, 2023. 26

[366] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. Adaptive multi-modalities fusion in sequential recommendation systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 843–853, 2023. 26

[367] Ning Wu, Ming Gong, Linjun Shou, Jian Pei, and Daxin Jiang. Ruel: Retrieval-augmented user representation with edge browser logs for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4871–4878, 2023. 26

[368] Hacer Turgut, Tan Doruk Yetki, Ömür Bali, and Tayfun Arda Yücel. Prod2vec-var: A session based recommendation system with enhanced diversity. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5253–5254, 2023. 26

[369] Ziyi Kou, Saurav Manchanda, Shih-Ting Lin, Min Xie, Haixun Wang, and Xiangliang Zhang. Modeling sequential collaborative user behaviors for seller-aware next basket recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1097–1106, 2023. 26, 34, 132

[370] Shutong Qiao, Wei Zhou, Junhao Wen, Hongyu Zhang, and Min Gao. Bi-channel multiple sparse graph attention networks for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2075–2084, 2023. 26

[371] Xin Xia, Junliang Yu, Guandong Xu, and Hongzhi Yin. Towards communication-efficient model updating for on-device session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2795–2804, 2023. 26

[372] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. Diffusion augmentation for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1576–1586, 2023. 26

[373] Dianer Yu, Qian Li, Hongzhi Yin, and Guandong Xu. Causality-guided graph learning for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3083–3093, 2023. 26

[374] Shigang Quan, Shui Liu, Zhenzhe Zheng, and Fan Wu. Enhancing repeat-aware recommendation from a temporal-sequential perspective. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2095–2105, 2023. 26

[375] Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. Attention calibration for transformer-based sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3595–3605, 2023. 26, 154

[376] Qian Chen, Jianjun Li, Zhiqiang Guo, Guohui Li, and Zhiying Deng. Attribute-enhanced dual channel representation learning for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3793–3797, 2023. 26

[377] Yu Hao Chou and Pu Jen Cheng. Incorporating co-purchase correlation for next-basket recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3823–3827, 2023. 26, 34, 132

[378] Zewen Long, Liang Wang, Qiang Liu, and Shu Wu. Personalized interest sustainability modeling for sequential poi recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4145–4149, 2023. 26

[379] Zhenlei Wang, Xu Chen, Rui Zhou, Quanyu Dai, Zhenhua Dong, and Ji-Rong Wen. Sequential recommendation with user causal behavior discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 28–40. IEEE, 2023. 26

[380] Zhikai Wang and Yanyan Shen. Incremental learning for multi-interest sequential recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1071–1083. IEEE, 2023. 26

[381] Huizi Wu, Hui Fang, Zhu Sun, Cong Geng, Xinyu Kong, and Yew-Soon Ong. A generic reinforced explainable framework with knowledge graph for session-based recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1260–1272. IEEE, 2023. 26

[382] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Guanfeng Liu, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. Contrastive enhanced slide filter mixer for sequential recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2673–2685. IEEE, 2023. 26

[383] Chi Zhang, Qilong Han, Rui Chen, Xiangyu Zhao, Peng Tang, and Hongtao Song. Ssdrec: Self-augmented sequence denoising for sequential recommendation. *arXiv preprint arXiv:2403.04278*, 2024. 26

[384] Yongjing Hao, Pengpeng Zhao, Junhua Fang, Jianfeng Qu, Guanfeng Liu, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. Meta-optimized joint generative and contrastive learning for sequential recommendation. *arXiv preprint arXiv:2310.13925*, 2023. 26

[385] Binquan Wu, Yu Cheng, Haitao Yuan, and Qianli Ma. When multi-behavior meets multi-interest: Multi-behavior sequential recommendation with multi-interest self-supervised learning. 26

[386] Zongzhi Han, Zhonghong Ou, Yifan Zhu, Xiaodong Li, and Meina Song. Fm-ignn: Interaction graph neural network with fine-grained matching for session-based recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 130–139. IEEE, 2023. 26

[387] Bingsen Huang, Jinwei Luo, Weihao Du, Weike Pan, and Zhong Ming. Cascaded cross attention for review-based sequential recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 170–179. IEEE, 2023. 26

[388] Xinyi Zhou, Jipeng Jin, Li Ma, Xiaofeng Gao, Jianbo Yang, Xiongwen Yang, and Lei Xiao. Script: Sequential cross-meta-information recommendation in pretrain and prompt paradigm. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 888–897. IEEE, 2023. 26, 83, 84

[389] Zhaohao Lin, Weike Pan, and Zhong Ming. Privacy-preserving cross-domain sequential recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1139–1144. IEEE, 2023. 26

[390] Longxiang Shi, Zilin Zhang, Shoujin Wang, Qi Zhang, Minghui Wu, Cheng Yang, and Shijian Li. Session-based interactive recommendation via deep reinforcement learning. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1319–1324. IEEE, 2023. 26

[391] Tianxing Wang, Can Wang, Hui Tian, and Hong Shen. Geomixer: The mlp-based sequential poi recommender with travel routing modelling. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1373–1378. IEEE, 2023. 26

[392] Chunjing Xiao, Yan Shen, Yuxiang Zhang, and Shijie Li. Graph collaborative optimization for sequential recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1427–1432. IEEE, 2023. 26

[393] Wanqi Xue, Qingpeng Cai, Ruohan Zhan, Dong Zheng, Peng Jiang, and Bo An. Resact: Reinforcing long-term engagement in sequential recommendation with residual actor (iclr'23). 2023. 26

[394] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*, 2024. 26

[395] Álvaro Labarca Silva, Denis Parra, and Rodrigo Toro Icarte. On the unexpected effectiveness of reinforcement learning for sequential recommendation. In *Forty-first International Conference on Machine Learning*. 26

[396] Huanhuan Yuan, Pengpeng Zhao, Xuefeng Xian, Guanfeng Liu, Victor S Sheng, and Lei Zhao. Sequential recommendation with probabilistic logical reasoning. *arXiv preprint arXiv:2304.11383*, 2023. 26, 34

[397] Yalin Yu, Enneng Yang, Guibing Guo, Linying Jiang, and Xingwei Wang. Basket representation learning by hypergraph convolution on repeated items for next-basket recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 2415–2422, 2023. 26, 34, 132, 182

[398] Huiyuan Chen, Kaixiong Zhou, Zhimeng Jiang, Chin-Chia Michael Yeh, Xiaoting Li, Menghai Pan, Yan Zheng, Xia Hu, and Hao Yang. Probabilistic masked attention networks for explainable sequential recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 2068–2076, 2023. 26, 34

[399] Lianyong Qi, Yuwen Liu, Weiming Liu, Shichao Pei, Xiaolong Xu, Xuyun Zhang, Yingjie Wang, and Wanchun Dou. Counterfactual user sequence synthesis augmented with continuous time dynamic preference modeling for sequential poi recommendation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024. 26, 34

[400] Liu Chong, Xiaoyang Liu, Rongqin Zheng, Lixin Zhang, Xiaobo Liang, Juntao Li, Lijun Wu, Min Zhang, and Leyu Lin. Ct4rec: Simple yet effective consistency training for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3901–3913, 2023. 26

[401] Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3434–3445, 2023. 26, 154

[402] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for

sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267, 2023. 26

[403] Hyunsoo Chung, AI Omnious, and Jungtaek Kim. Leveraging uniformity of normalized embeddings for sequential recommendation. *History*, 4(5):6. 26

[404] Yatong Sun, Bin Wang, Zhu Sun, Xiaochun Yang, and Yan Wang. Theoretically guaranteed bidirectional data rectification for robust sequential recommendation. *Advances in Neural Information Processing Systems*, 36, 2024. 26

[405] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36, 2024. 26

[406] Xin Liu, Zheng Li, Yifan Gao, Jingfeng Yang, Tianyu Cao, Zhengyang Wang, Bing Yin, and Yangqiu Song. Enhancing user intent capture in session-based recommendation with attribute patterns. *Advances in Neural Information Processing Systems*, 36, 2024. 26

[407] Sirui Wang, Peiguang Li, Yunsen Xian, and Hongzhi Zhang. Beyond the sequence: Statistics-driven pre-training for stabilizing sequential recommendation model. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 723–729, 2023. 26

[408] Vivian Lai, Huiyuan Chen, Chin-Chia Michael Yeh, Minghua Xu, Yiwei Cai, and Hao Yang. Enhancing transformers without self-supervised learning: A loss landscape perspective in sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 791–797, 2023. 26

[409] Wenqi Sun, Ruobing Xie, Junjie Zhang, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. Generative next-basket recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 737–743, 2023. 26, 34, 132

[410] Mostafa Rahmani, James Caverlee, and Fei Wang. Incorporating time in sequential recommendation models. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 784–790, 2023. 26, 155

[411] Marta Moscati, Christian Wallmann, Markus Reiter-Haas, Dominik Kowald, Elisabeth Lex, and Markus Schedl. Integrating the act-r framework with collaborative filtering for explainable sequential music recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 840–847, 2023. 26

[412] Yaming Yang, Jieyu Zhang, Yujing Wang, Zheng Miao, and Yunhai Tong. Multiple connectivity views for session-based recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1000–1006, 2023. 26

[413] Aayush Singha Roy, Edoardo D'Amico, Elias Tragos, Aonghus Lawlor, and Neil Hurley. Scalable deep q-learning for session-based slate recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 877–882, 2023. 26

[414] Aleksey Romanov, Oleg Lashinin, Marina Ananyeva, and Sergey Kolesnikov. Time-aware item weighting for the next basket recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 985–992, 2023. 26, 34, 132, 182

[415] Alejandro Ariza-Casabona, Maria Salamó, Ludovico Boratto, and Gianni Fenu. Towards self-explaining sequence-aware recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 904–911, 2023. 26

[416] Zitao Xu, Weike Pan, and Zhong Ming. A multi-view graph contrastive learning framework for cross-domain sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 491–501, 2023. 26

[417] Yichi Zhang, Guisheng Yin, and Yuxin Dong. Contrastive learning with frequency-domain interest trends for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 141–150, 2023. 26

[418] Haibo Liu, Zhixiang Deng, Liang Wang, Jinjia Peng, and Shi Feng. Distribution-based learnable filters with side information for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 78–88, 2023. 26

[419] Xiaoxin Ye, Yun Li, and Lina Yao. Dream: Decoupled representation via extraction attention module and supervised contrastive learning for cross-domain sequential recommender. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 479–490, 2023. 26

[420] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, Jaeboum Kim, Shoujin Wang, and Sunghun Kim. Equivariant contrastive learning for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 129–140, 2023. 26

[421] Aleksandr Vladimirovich Petrov and Craig Macdonald. gsasrec: Reducing overconfidence in sequential recommendation trained with negative sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 116–128, 2023. 26

[422] Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten de Rijke. Masked and swapped sequence modeling for next novel basket recommendation in grocery shopping. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 35–46, 2023. 26, 34, 132

[423] Bowen Zheng, Yupeng Hou, Wayne Xin Zhao, Yang Song, and Hengshu Zhu. Reciprocal sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 89–100, 2023. 26

[424] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. Spare: Shortest path global item relations for efficient session-based recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 58–69, 2023. 26

[425] Chengxi Li, Yejing Wang, Qidong Liu, Xiangyu Zhao, Wanyu Wang, Yiqi Wang, Lixin Zou, Wenqi Fan, and Qing Li. Strec: Sparse transformer for sequential recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 101–111, 2023. 26, 155

[426] Yunzhu Pan, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Depeng Jin, and Yong Li. Understanding and modeling passive-negative feedback for short-video sequential recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 540–550, 2023. 26

[427] Xumei Xi, Yuke Zhao, Quan Liu, Liwen Ouyang, and Yang Wu. Integrating offline reinforcement learning with transformers for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1–1, 2023. 26, 154

[428] Kai Wang, Zhene Zou, Minghao Zhao, Qilin Deng, Yue Shang, Yile Liang, Runze Wu, Xudong Shen, Tangjie Lyu, and Changjie Fan. Rl4rs: A real-world dataset for reinforcement learning based recommender system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2935–2944, 2023. 26

[429] Yanling Wang, Yuchen Liu, Qian Wang, Cong Wang, and Chenliang Li. Poisoning self-supervised learning based sequential recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 300–310, 2023. 26

[430] Zhiying Deng, Jianjun Li, Zhiqiang Guo, Wei Liu, Li Zou, and Guohui Li. Multi-view multi-aspect neural networks for next-basket recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1283–1292, 2023. 26, 34, 133

[431] Qian Chen, Zhiqiang Guo, Jianjun Li, and Guohui Li. Knowledge-enhanced multi-view graph neural networks for session-based recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 352–361, 2023. 26

[432] Kibum Kim, Dongmin Hyun, Sukwon Yun, and Chanyoung Park. Melt: Mutual enhancement of long-tailed user and item for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on Research and development in information retrieval*, pages 68–77, 2023. 26

[433] Rui Zhou, Xian Wu, Zhaopeng Qiu, Yefeng Zheng, and Xu Chen. Distributionally robust sequential recommnedation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–288, 2023. 26

[434] Heeyoon Yang, YunSeok Choi, Gahyung Kim, and Jee-Hyong Lee. Loam: improving long-tail session-based recommendation via niche walk augmentation and tail session mixup. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 527–536, 2023. 26

[435] Ran Li, Liang Zhang, Guannan Liu, and Junjie Wu. Next basket recommendation with intent-aware hypergraph adversarial network. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1303–1312, 2023. 26, 34, 133

[436] Hanwen Du, Huanhuan Yuan, Pengpeng Zhao, Fuzhen Zhuang, Guanfeng Liu, Lei Zhao, Yanchi Liu, and Victor S Sheng. Ensemble modeling with contrastive knowledge distillation for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 58–67, 2023. 26

[437] Langming Liu, Liu Cai, Chi Zhang, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Yifu Lv, Wenqi Fan, Yiqi Wang, Ming He, et al. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 289–299, 2023. 26

[438] Yaowen Ye, Lianghao Xia, and Chao Huang. Graph masked autoencoder for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–330, 2023. 26

[439] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor Sheng. Meta-optimized contrastive learning for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 89–98, 2023. 26

[440] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. Dual contrastive transformer for hierarchical preference modeling in sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 99–109, 2023. 26, 154

[441] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. A generic learning framework for sequential recommendation with distribution shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 331–340, 2023. 26

[442] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. Frequency enhanced hybrid attention network for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 78–88, 2023. 26

[443] Kai Ouyang, Xianghong Xu, Miaoxin Chen, Zuotong Xie, Hai-Tao Zheng, Shuangyong Song, and Yu Zhao. Mining interest trends and adaptively assigning sample weight for session-based recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2174–2178, 2023. 26

[444] Lu Fan, Jiashu Pu, Rongsheng Zhang, and Xiao-Ming Wu. Neighborhood-based hard negative mining for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2042–2046, 2023. 26

[445] Viet Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. Attention mixtures for time-aware sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1821–1826, 2023. 26

[446] Yulong Huang, Yang Zhang, Qifan Wang, Chenxu Wang, and Fuli Feng. Prediction then correction: An abductive prediction correction method for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2272–2276, 2023. 26

[447] Xiangkui Lu, Jun Wu, and Jianbo Yuan. Optimizing reciprocal rank with bayesian average for improved next item recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2236–2240, 2023. 26

[448] Ming Li, Yuanna Liu, Sami Jullien, Mozhdeh Ariannezhad, Mohammad Aliannejadi, Andrew Yates, and Maarten de Rijke. Are we really achieving better beyond-accuracy performance in next basket recommendation? *arXiv preprint arXiv:2405.01143*, 2024. 26, 34, 133

[449] Xiaokun Zhang, Bo Xu, Youlin Wu, Yuan Zhong, Hongfei Lin, and Fenglong Ma. Finerec: Exploring fine-grained sequential recommendation. *arXiv preprint arXiv:2404.12975*, 2024. 26

[450] Ziru Liu, Shuchang Liu, Zijian Zhang, Qingpeng Cai, Xiangyu Zhao, Kesen Zhao, Lantao Hu, Peng Jiang, and Kun Gai. Sequential recommendation for optimizing both immediate feedback and long-term retention. *arXiv preprint arXiv:2404.03637*, 2024. 26

[451] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. Towards llm-recsys alignment with textual id learning. *arXiv preprint arXiv:2403.19021*, 2024. 26

[452] Minjin Choi, Hye-young Kim, Hyunsouk Cho, and Jongwuk Lee. Multi-intent-aware session-based recommendation. *arXiv preprint arXiv:2405.00986*, 2024. 26

[453] Kaike Zhang, Qi Cao, Yunfan Wu, Fei Sun, Huawei Shen, and Xueqi Cheng. Lorec: Large language model for robust sequential recommendation against poisoning attacks. *arXiv preprint arXiv:2401.17723*, 2024. 26, 83, 85

[454] Xiaokun Zhang, Bo Xu, Zhaochun Ren, Xiaochen Wang, Hongfei Lin, and Fenglong Ma. Disentangling id and modality effects for session-based recommendation. *arXiv preprint arXiv:2404.12969*, 2024. 26

[455] Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. Sequential recommendation with latent relations based on large language model. *arXiv preprint arXiv:2403.18348*, 2024. 26, 83

[456] Xiaolong Xu, Hongsheng Dong, Lianyong Qi, Xuyun Zhang, Haolong Xiang, Xiaoyu Xia, Yanwei Xu, and Wanchun Dou. Cmclrec: cross-modal contrastive learning for user cold-start sequential recommendation. In *Proceeding of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024. 26

[457] Raksha Jalan, Tushar Prakash, and Niranjan Pedanekar. LLM-BRec: Personalizing session-based social recommendation with LLM-BERT fusion framework. In *The Second Workshop on Generative Information Retrieval*, 2024. URL `https://openreview.net/forum?id=gwHVlTNKsG`. 26

[458] Carles Balsells Rodas, Fan Yang, Zhishen Huang, and Yan Gao. Explainable uncertainty attribution for sequential recommendation. 2024. 26

[459] Li Nian, Ban Xin, Ling Cheng, Gao Chen, Hu Lantao, Jiang Peng, Gai Kun, Li Yong, and Liao Qingmin. Modeling user fatigue for sequential recommendation. *arXiv preprint arXiv:2405.11764*, 2024. 26

[460] Junchen Fu, Xuri Ge, Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, Jie Wang, and Joemon M Jose. Iisan: Efficiently adapting multimodal representation for sequential recommendation with decoupled peft. *arXiv preprint arXiv:2404.02059*, 2024. 26

[461] Yuxi Liu, Lianghao Xia, and Chao Huang. Selfgnn: Self-supervised graph neural networks for sequential recommendation. *arXiv preprint arXiv:2405.20878*, 2024. 26

[462] Huimin Zeng, Zhankui He, Zhenrui Yue, Julian McAuley, and Dong Wang. Fair sequential recommendation without user demographics. 2024. 26

[463] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795, 2024. 26

[464] Jing Xiao, Weike Pan, and Zhong Ming. A generic behavior-aware data augmentation framework for sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1578–1588, 2024. 26

[465] Yankun Ren, Zhongde Chen, Xinxing Yang, Longfei Li, Cong Jiang, Lei Cheng, Bo Zhang, Linjian Mo, and Jun Zhou. Enhancing sequential recommenders with augmented knowledge from aligned large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 345–354, 2024. 26

[466] Chung Park, Taesan Kim, Hyungjun Yoon, Junui Hong, Yelim Yu, Mincheol Cho, Minsung Choi, and Jaegul Choo. Pacer and runner: Cooperative learning framework between single-and cross-domain sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2071–2080, 2024. 26

[467] Pablo Zivic, Hernan Vazquez, and Jorge Sánchez. Scaling sequential recommendation models with transformers. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1567–1577, 2024. 26

[468] Muskan Gupta, Priyanka Gupta, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. Scm4sr: Structural causal model-based data augmentation for robust session-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2609–2613, 2024. 26

[469] Zhu Sun, Hongyang Liu, Xinghua Qu, Kaidong Feng, Yan Wang, and Yew Soon Ong. Large language models for intent-driven session recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–334, 2024. 26

[470] Junyoung Hwang, Hyunjun Ju, SeongKu Kang, Sanghwan Jang, and Hwanjo Yu. Multi-domain sequential recommendation via domain space learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2134–2144, 2024. 26

[471] Yi Yu, Kazunari Sugiyama, and Adam Jatowt. Sequential recommendation with collaborative explanation via mutual information maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1062–1072, 2024. 26

[472] Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 411–419, 2023. 26, 34, 153

[473] Jing Du, Zesheng Ye, Bin Guo, Zhiwen Yu, and Lina Yao. Idnp: Interest dynamics modeling using generative neural processes for sequential recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 481–489, 2023. 26

[474] Zihao Li, Xianzhi Wang, Chao Yang, Lina Yao, Julian McAuley, and Guandong Xu. Exploiting explicit and implicit item relationships for session-based recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 553–561, 2023. 26, 73, 134

[475] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 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*, pages 168–176, 2023. 26, 66

[476] Hengchang Hu, Wei Guo, Xu Liu, Yong Liu, Ruiming Tang, Rui Zhang, and Min-Yen Kan. User behavior enriched temporal knowledge graphs for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 266–275, 2024. 26, 75, 183

[477] Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 930–938, 2024. 26, 62

[478] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Guanfeng Liu, Fuzhen Zhuang, and Victor S Sheng. Intent contrastive learning with cross subsequences for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 548–556, 2024. 26, 81

[479] Tianyu Zhu, Yansong Shi, Yuan Zhang, Yihong Wu, Fengran Mo, and Jian-Yun Nie. Collaboration and transition: Distilling item transitions into multi-query self-attention for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1003–1011, 2024. 26, 70

[480] Guanyu Lin, Chen Gao, Yu Zheng, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Zhiheng Li, Depeng Jin, Yong Li, et al. Mixed attention network for cross-domain sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 405–413, 2024. 26, 66

[481] Aleksandr V Petrov and Craig Macdonald. Recjpq: training large-catalogue sequential recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 538–547, 2024. 26

[482] Xin Xin, Liu Yang, Ziqi Zhao, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. On the effectiveness of unlearning in session-based recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 855–863, 2024. 26

[483] Zhichao Feng, Pengfei Wang, Kaiyuan Li, Chenliang Li, and Shangguang Wang. Contextual mab oriented embedding denoising for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 199–207, 2024. 26

[484] Haw-Shiuan Chang, Nikhil Agarwal, and Andrew McCallum. To copy, or not to copy; that is a critical issue of the output softmax layer in neural sequential recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 67–76, 2024. 26

[485] Xiaolin Lin, Jinwei Luo, Junwei Pan, Weike Pan, Zhong Ming, Xun Liu, Shudong Huang, and Jie Jiang. Multi-sequence attentive user representation learning for side-information integrated sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 414–423, 2024. 26, 66, 81

[486] Jiyuan Yang, Yue Ding, Yidan Wang, Pengjie Ren, Zhumin Chen, Fei Cai, Jun Ma, Rui Zhang, Zhaochun Ren, and Xin Xin. Debiasing sequential recommenders through distributionally robust optimization over system exposure. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 882–890, 2024. 26

[487] Xuewei Li, Hongwei Chen, Jian Yu, Mankun Zhao, Tianyi Xu, Wenbin Zhang, and Mei Yu. Global heterogeneous graph and target interest denoising for multi-behavior sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 387–395, 2024. 26, 64

[488] Yujie Lin, Chenyang Wang, Zhumin Chen, Zhaochun Ren, Xin Xin, Qiang Yan, Maarten de Rijke, Xiuzhen Cheng, and Pengjie Ren. A self-correcting sequential recommender. In *Proceedings of the ACM Web Conference 2023*, pages 1283–1293, 2023. 26, 34

[489] Guanyu Lin, Chen Gao, Yu Zheng, Jianxin Chang, Yanan Niu, Yang Song, Zhiheng Li, Depeng Jin, and Yong Li. Dual-interest factorization-heads attention for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 917–927, 2023. 26, 34, 66

[490] Wenzhuo Song, Shoujin Wang, Yan Wang, Kunpeng Liu, Xueyan Liu, and Minghao Yin. A counterfactual collaborative session-based recommender system. In *Proceedings of the ACM Web Conference 2023*, pages 971–982, 2023. 26, 34

[491] Ziwei Fan, Zhiwei Liu, Hao Peng, and Philip S Yu. Mutual wasserstein discrepancy minimization for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1375–1385, 2023. 26, 34

[492] Jiajie Su, Chaochao Chen, Weiming Liu, Fei Wu, Xiaolin Zheng, and Haoming Lyu. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 165–176, 2023. 26, 34

[493] Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiajie Su, Xinting Liao, Mengling Hu, and Yanchao Tan. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 383–394, 2023. 26, 34

[494] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. Dual intent enhanced graph neural network for session-based new item recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 684–693, 2023. 26, 34

[495] Lingfeng Shi, Yuang Liu, Jun Wang, and Wei Zhang. Quantize sequential recommenders without private data. In *Proceedings of the ACM Web Conference 2023*, pages 1043–1052, 2023. 26

[496] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023*, pages 1063–1073, 2023. 26

[497] Jiahao Liang, Xiangyu Zhao, Muyang Li, Zijian Zhang, Wanyu Wang, Haochen Liu, and Zitao Liu. Mmmlp: multi-modal multilayer perceptron for sequential recommendations. In *Proceedings of the ACM Web Conference 2023*, pages 1109–1117, 2023. 26

[498] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pages 1162–1171, 2023. 26

[499] Muyang Li, Zijian Zhang, Xiangyu Zhao, Wanyu Wang, Minghao Zhao, Runze Wu, and Ruocheng Guo. Automlp: Automated mlp for sequential recommendations. In *Proceedings of the ACM Web Conference 2023*, pages 1190–1198, 2023. 26, 60

[500] Zhihao Zhu, Chenwang Wu, Rui Fan, Defu Lian, and Enhong Chen. Membership inference attacks against sequential recommender systems. In *Proceedings of the ACM Web Conference 2023*, pages 1208–1219, 2023. 26

[501] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. Modeling temporal positive and negative excitation for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1252–1263, 2023. 26

[502] Zheqi Lv, Wenqiao Zhang, Zhengyu Chen, Shengyu Zhang, and Kun Kuang. Intelligent model update strategy for sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3117–3128, 2024. 26, 34

[503] Yongqiang Han, Hao Wang, Kefan Wang, Likang Wu, Zhi Li, Wei Guo, Yong Liu, Defu Lian, and Enhong Chen. Efficient noise-decoupling for multi-behavior sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3297–3306, 2024. 26, 34, 81

[504] Peilin Zhou, You-Liang Huang, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, and Sunghun Kim. Is contrastive learning necessary? a study of data augmentation vs contrastive learning in sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3854–3863, 2024. 26, 34, 81

[505] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM on Web Conference 2024*, pages 3876–3887, 2024. 26, 34, 84

[506] Wujiang Xu, Qitian Wu, Runzhong Wang, Mingming Ha, Qiongxu Ma, Linxun Chen, Bing Han, and Junchi Yan. Rethinking cross-domain sequential recommendation under open-world assumptions. In *Proceedings of the ACM on Web Conference 2024*, pages 3173–3184, 2024. 26, 34

[507] Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3207–3216, 2024. 26, 34, 83

[508] Jiayi Xie, Shang Liu, Gao Cong, and Zhenzhong Chen. Unifiedssr: A unified framework of sequential search and recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3410–3419, 2024. 26, 34, 81

[509] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3497–3508, 2024. 26, 34, 83, 84

[510] Shaowei Wei, Zhengwei Wu, Xin Li, Qintong Wu, Zhiqiang Zhang, Jun Zhou, Lihong Gu, and Jinjie Gu. Leave no one behind: Online self-supervised self-distillation for sequential recommendation. *arXiv preprint arXiv:2404.07219*, 2024. 26, 34, 81

[511] Yongjing Hao, Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Guanfeng Liu, and Xiaofang Zhou. Feature-level deeper self-attention network with contrastive learning for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[512] Hanwen Du, Huanhuan Yuan, Pengpeng Zhao, Deqing Wang, Victor S Sheng, Yanchi Liu, Guanfeng Liu, and Lei Zhao. Feature-aware contrastive learning with bidirectional transformers for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[513] Hongzhi Liu, Yao Zhu, and Zhonghai Wu. Knowledge graph-based behavior denoising and preference learning for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[514] Ziyang Wang, Wei Wei, Shanshan Feng, Xian-Ling Mao, Minghui Qiu, Dangyang Chen, and Rui Fang. Exploiting group-level behavior pattern for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[515] Ting-Ting Su, Chang-Dong Wang, Wu-Dong Xi, Jian-Huang Lai, and S Yu Philip. Hierarchical alignment with polar contrastive learning for next-basket recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34, 133

[516] Pan Li, Maofei Que, and Alexander Tuzhilin. Dual contrastive learning for efficient static feature representation in sequential recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[517] Xiaokun Zhang, Bo Xu, Fenglong Ma, Chenliang Li, Liang Yang, and Hongfei Lin. Beyond co-occurrence: Multi-modal session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26, 34

[518] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Ticoserec: Augmenting data to uniform sequences by time intervals for effective recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26

[519] Yicong Li, Hongxu Chen, Yile Li, Lin Li, S Yu Philip, and Guandong Xu. Reinforcement learning based path exploration for sequential explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 26

[520] Zhikai Wang and Yanyan Shen. A framework for elastic adaptation of user multiple intents in sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26, 34

[521] Kang Yang, Ruiyun Yu, Bingyang Guo, and Jie Li. Interaction subgraph sequential topology-aware network for transferable recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26, 34

[522] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26, 34, 156

[523] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. Personalized prompt for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26, 34, 83, 84, 85

[524] Zihan Liao, Xiaodong Wu, Shuo Shang, Jun Wang, and Wei Zhang. Modeling dynamic item tendency bias in sequential recommendation with causal intervention. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26

[525] Chunnan Wang, Hongzhi Wang, Junzhe Wang, and Guosheng Feng. Autosr: Automatic sequential recommendation system design. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 26

[526] Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42 (3):1–28, 2023. 26

[527] Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. A next basket recommendation reality check. *ACM Transactions on Information Systems*, 41(4):1–29, 2023. 26, 34, 133

[528] Kun Zhou, Hui Wang, Ji-rong Wen, and Wayne Xin Zhao. Enhancing multi-view smoothness for sequential recommendation models. *ACM Transactions on Information Systems*, 41(4):1–27, 2023. 26

[529] Xin Xia, Junliang Yu, Qinyong Wang, Chaoqun Yang, Nguyen Quoc Viet Hung, and Hongzhi Yin. Efficient on-device session-based recommendation. *ACM Transactions on Information Systems*, 2023. 26

[530] Lyuxin Xue, Deqing Yang, Shuoyao Zhai, Yuxin Li, and Yanghua Xiao. Learning dual-view user representations for enhanced sequential recommendation. *ACM Transactions on Information Systems*, 41(4):1–26, 2023. 26

[531] Muyang Ma, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Huasheng Liang, Jun Ma, and Maarten De Rijke. Improving transformer-based sequential recommenders through preference editing. *ACM Transactions on Information Systems*, 41(3):1–24, 2023. 26, 155

[532] Naicheng Guo, Xiaolei Liu, Shaoshuai Li, Qiongxu Ma, Kaixin Gao, Bing Han, Lin Zheng, Sheng Guo, and Xiaobo Guo. Poincaré heterogeneous graph neural networks for sequential recommendation. *ACM Transactions on Information Systems*, 41(3):1–26, 2023. 26

[533] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Sequential recommendation with multiple contrast signals. *ACM Transactions on Information Systems*, 41(1):1–27, 2023. 26, 34, 153

[534] Liwei Huang, Yutao Ma, Yanbo Liu, Bohong Danny Du, Shuliang Wang, and Deyi Li. Position-enhanced and time-aware graph convolutional network for sequential recommendations. *ACM Transactions on Information Systems*, 41 (1):1–32, 2023. 26, 155

[535] Surong Yan, Chenglong Shi, Haosen Wang, Lei Chen, Ling Jiang, Ruilin Guo, and Kwei-Jay Lin. Teach and explore: A multiplex information-guided effective and efficient reinforcement learning for sequential recommendation. *ACM Transactions on Information Systems*, 2023. 26

[536] Tianzi Zang, Yanmin Zhu, Ruohan Zhang, Chunyang Wang, Ke Wang, and Jiadi Yu. Contrastive multi-view interest learning for cross-domain sequential recommendation. *ACM Transactions on Information Systems*, 42(3):1–30, 2023. 26

[537] Xiaokun Zhang, Bo Xu, Fenglong Ma, Chenliang Li, Yuan Lin, and Hongfei Lin. Bi-preference learning heterogeneous hypergraph networks for session-based recommendation. *ACM Transactions on Information Systems*, 42(3): 1–28, 2023. 26

[538] Zhizhuo Yin, Kai Han, Pengzi Wang, and Xi Zhu. H3gnn: Hybrid hierarchical hypergraph neural network for personalized session-based recommendation. *ACM Transactions on Information Systems*, 42(3):1–30, 2023. 26

[539] Zhongwei Wan, Xin Liu, Benyou Wang, Jiezhong Qiu, Boyu Li, Ting Guo, Guangyong Chen, and Yang Wang. Spatio-temporal contrastive learning-enhanced gnns for session-based recommendation. *ACM Transactions on Information Systems*, 42(2):1–26, 2023. 26

[540] Yatong Sun, Xiaochun Yang, Zhu Sun, and Bin Wang. Berd+: A generic sequential recommendation framework by eliminating unreliable data with item-and attribute-level signals. *ACM Transactions on Information Systems*, 42(2):1–33, 2023. 26

[541] Qing Yin, Hui Fang, Zhu Sun, and Yew-Soon Ong. Understanding diversity in session-based recommendation. *ACM Transactions on Information Systems*, 42(1):1–34, 2023. 26

[542] Dongjing Wang, Xin Zhang, Yuyu Yin, Dongjin Yu, Guandong Xu, and Shuiguang Deng. Multi-view enhanced graph attention network for session-based music recommendation. *ACM Transactions on Information Systems*, 42(1):1–30, 2023. 26

[543] Yao Wu, Jian Cao, and Guandong Xu. Faster: A dynamic fairness-assurance strategy for session-based recommender systems. *ACM Transactions on Information Systems*, 42(1):1–26, 2023. 26

[544] Xingrui Zhuo, Shengsheng Qian, Jun Hu, Fuxin Dai, Kangyi Lin, and Gongqing Wu. Multi-hop multi-view memory transformer for session-based recommendation. *ACM Transactions on Information Systems*, 2024. 26

[545] Chaoyu Shi, Pengjie Ren, Dongjie Fu, Xin Xin, Shansong Yang, Fei Cai, Zhaochun Ren, and Zhumin Chen. Diversifying sequential recommendation with retrospective and prospective transformers. *ACM Transactions on Information Systems*, 2024. 26, 155

[546] Yabin Zhang, Zhenlei Wang, Wenhui Yu, Lantao Hu, Peng Jiang, Kun Gai, and Xu Chen. Soft contrastive sequential recommendation. *ACM Transactions on Information Systems*, 2024. 26

[547] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. Smlp4rec: An efficient all-mlp architecture for sequential recommendations. *ACM Transactions on Information Systems*, 42(3):1–23, 2024. 26

[548] Xin Wang, Hong Chen, Zirui Pan, Yuwei Zhou, Chaoyu Guan, Lifeng Sun, and Wenwu Zhu. Automated disentangled sequential recommendation with large language models. *ACM Transactions on Information Systems*. 26

[549] Surong Yan, Chenglong Shi, Haosen Wang, Lei Chen, Ling Jiang, Ruilin Guo, and Kwei-Jay Lin. Teach and explore: A multiplex information-guided effective and efficient reinforcement learning for sequential recommendation. *ACM Transactions on Information Systems*, 42(5):1–26, 2024. 26

[550] Yiheng Jiang, Yuanbo Xu, Yongjian Yang, Funing Yang, Pengyang Wang, Chaozhuo Li, Fuzhen Zhuang, and Hui Xiong. Trimlp: A foundational mlp-like architecture for sequential recommendation. *ACM Transactions on Information Systems*. 26

[551] Tianze Luo, Yong Liu, and Sinno Jialin Pan. Collaborative sequential recommendations via multi-view gnn-transformers. *ACM Transactions on Information Systems*, 2024. 26

[552] Shu Chen, Zitao Xu, Weike Pan, Qiang Yang, and Zhong Ming. A survey on cross-domain sequential recommendation. *arXiv preprint arXiv:2401.04971*, 2024. 28

[553] Sara Latifi, Dietmar Jannach, and Andrés Ferraro. Sequential recommendation: A study on transformers, nearest neighbors and sampled metrics. *Information Sciences*, 609:660–678, 2022. 28

[554] Shoujin Wang, Qi Zhang, Liang Hu, Xiuzhen Zhang, Yan Wang, and Charu Aggarwal. Sequential/session-based recommendations: Challenges, approaches, applications and opportunities. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3425–3428, 2022. 28

[555] Shoujin Wang, Longbing Cao, and Yan Wang. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*, 2019. 28

[556] Le Yu, Leilei Sun, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. Predicting temporal sets with deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1083–1091, 2020. 34, 178, 181

[557] Le Yu, Zihang Liu, Tongyu Zhu, Leilei Sun, Bowen Du, and Weifeng Lv. Predicting temporal sets with simplified fully connected networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4835–4844, 2023. 34, 181

[558] Seungjae Jung, Young-Jin Park, Jisu Jeong, Kyung-Min Kim, Hiun Kim, Minkyu Kim, and Hanock Kwak. Global-local item embedding for temporal set prediction. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 674–679, 2021. 34, 181

[559] Nannan Shi, Le Yu, Leilei Sun, Lihua Wang, Chunming Lin, and Ruixing Zhang. Deep heterogeneous network for temporal set prediction. *Knowledge-Based Systems*, 223:107039, 2021. 34, 181

[560] Le Yu, Guanghui Wu, Leilei Sun, Bowen Du, and Weifeng Lv. Element-guided temporal graph representation learning for temporal sets prediction. In *Proceedings of the ACM Web Conference 2022*, pages 1902–1913, 2022. 34, 181

[561] Le Yu, Zihang Liu, Leilei Sun, Bowen Du, Chuanren Liu, and Weifeng Lv. Continuous-time user preference modelling for temporal sets prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 34, 181

[562] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018. 43

[563] Jiajin Wu, Bo Yang, Runze Mao, and Qing Li. Popularity-aware sequential recommendation with user desire. *Expert Systems with Applications*, 237: 121429, 2024. 45

[564] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017. 46

[565] Ghim-Eng Yap, Xiao-Li Li, and Philip S Yu. Effective next-items recommendation via personalized sequential pattern mining. In *Database Systems for Advanced Applications: 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part II 17*, pages 48–64. Springer, 2012. 46

[566] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993. 46

[567] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000. 46

[568] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering*, pages 215–224. IEEE Piscataway, NJ, USA, 2001. 46

[569] Niti Ashishkumar Desai and Amit Ganatra. Buying scenario and recommendation of purchase by constraint based sequential pattern mining from time stamp based sequential dataset. *Procedia Computer Science*, 45:166–175, 2015. 47

[570] R Bharathi, A Juliet, and ML Valarmathi. Mining sequential pattern for online shopping recommendation system using prefix span algorithm. In *AIP Conference Proceedings*, volume 2587. AIP Publishing, 2023. 48

[571] AA Markov. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. dynamic probabilistic systems, volume 1: Markov chains, 1971. 48

[572] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970. 49

[573] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. 50

[574] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Handbook for Automatic Computation: Volume II: Linear Algebra*, pages 134–151. Springer, 1971. 51

[575] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007. 51

[576] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009. 51, 88, 115, 141

[577] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887, 2008. 51

[578] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008. 51, 112

[579] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999. 51

[580] Steffen Rendle. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE, 2010. 51, 55

[581] Sepp Hochreiter and Júrgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 61, 109

[582] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 61, 109

[583] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 63

[584] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 64

[585] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 64, 149

[586] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 64

[587] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 64

[588] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297, 2016. 64

[589] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 64

[590] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016. 64, 66

[591] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 64, 71

[592] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, Qiushi Huang, Jintao Li, and Tao Mei. Sequential prediction of social media popularity with deep temporal context networks. *arXiv preprint arXiv:1712.04443*, 2017. 65

[593] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 68

[594] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2671–2679, 2019. 68

[595] *Csan: Contextual self-attention network for user sequential recommendation*, 2018. 69

[596] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pages 1–4, 2019. 69

[597] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. Sdm: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2635–2643, 2019. 69

[598] Ming Zhang, Shixing Liu, and Yifan Wang. Str-sa: Session-based thread recommendation for online course forum with self-attention. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 374–381. IEEE, 2020. 69

[599] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018. 72

[600] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 72

[601] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 72

[602] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 72, 131

[603] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. 72

[604] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020. 74

[605] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013. 75

[606] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 76

[607] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 77

[608] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 77

[609] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011. 78

[610] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 78

[611] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 78

[612] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 81

[613] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 81

[614] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021. 81

[615] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 83

[616] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 83

[617] Anthropic. Claude 3, 2024. URL `https://www.anthropic.com/claude`. 83

[618] Imtiaz Ahmed and Raisa Islam. Gemini-the most powerful llm: Myth or truth. *Authorea Preprints*, 2024. 83

[619] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 83

[620] Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, et al. Reka core, flash, and edge: A series of powerful multimodal language models. *arXiv preprint arXiv:2404.12387*, 2024. 83

[621] Qwen2 technical report. 2024. 83

[622] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 84

[623] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 84

[624] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 84

[625] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017. 86, 87

[626] Shyan S Chen and Leon Greenberg. Hyperbolic spaces. In *Contributions to analysis*, pages 49–87. Elsevier, 1974. 87

[627] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, 2008. 88

[628] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007. 88

[629] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010. 89

[630] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993. 90

[631] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 90, 116, 168, 197

[632] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*, pages 1–9, 2014. 99

[633] Xavier Amatriain and Justin Basilico. Recommender systems in industry: A netflix case study. In *Recommender systems handbook*, pages 385–419. Springer, 2015. 99

[634] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016. 99

[635] Sahin Cem Geyik, Qi Guo, Bo Hu, Cagri Ozcaglar, Ketan Thakkar, Xianren Wu, and Krishnaram Kenthapadi. Talent search and recommendation systems at linkedin: Practical challenges and lessons learned. In *The 41st*

International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 1353–1354, 2018. 99, 100

[636] Chaoyue He, Yong Liu, Qingyu Guo, and Chunyan Miao. Multi-scale quasi-rnn for next item recommendation. *arXiv preprint arXiv:1902.09849*, 2019. 103

[637] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 108

[638] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4580–4584. IEEE, 2015. 108

[639] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015. 108

[640] Chenglong Wang, Feijun Jiang, and Hongxia Yang. A hybrid framework for text modeling with convolutional rnn. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2061–2069, 2017. 108

[641] Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar. Improving cnn-rnn hybrid networks for handwriting recognition. In *2018 16th international conference on frontiers in handwriting recognition (ICFHR)*, pages 80–85. IEEE, 2018. 108

[642] Aditya Khamparia, Babita Pandey, Shrasti Tiwari, Deepak Gupta, Ashish Khanna, and Joel JPC Rodrigues. An integrated hybrid cnn–rnn model for visual description and generation of captions. *Circuits, Systems, and Signal Processing*, 39:776–788, 2020. 108

[643] Mohsin Ashraf, Fazeel Abid, Ikram Ud Din, Jawad Rasheed, Mirsat Yesiltepe, Sook Fern Yeo, and Merve T Ersoy. A hybrid cnn and rnn variant model for music classification. *Applied Sciences*, 13(3):1476, 2023. 109

[644] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*, 2016. 109

[645] Tao Lei, Yu Zhang, and Yoav Artzi. Training rnns as fast as cnns. *arXiv preprint 1709.02755*, 2017. 109

[646] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Recurrently controlled recurrent networks. *Advances in neural information processing systems*, 31, 2018. 109

[647] Qingchun Bai, Jie Zhou, and Liang He. Pg-rnn: using position-gated recurrent neural networks for aspect-based sentiment classification. *The Journal of Supercomputing*, 78(3):4073–4094, 2022. 109

[648] Moritz Wolter and Angela Yao. Complex gated recurrent neural networks. *Advances in neural information processing systems*, 31, 2018. 109

[649] Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljacic, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. *Neural computation*, 31(4):765–783, 2019. 109

[650] Alan Said and Alejandro Bellogín. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 129–136, 2014. 112

[651] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017. 112, 114, 116, 140, 141, 143, 164, 194

[652] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. Next item recommendation with self-attentive metric learning. In *Thirty-Third AAAI Conference on Artificial Intelligence*, volume 9, 2019. 114, 141, 164, 194

[653] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*, pages 729–739, 2018. 114, 140, 164, 194

[654] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, volume 15, pages 2069–2075, 2015. 114, 141, 164, 194

[655] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 161–169. ACM, 2017. 114, 141, 164, 194

[656] 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. 116, 168, 197

[657] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 116, 168, 197

[658] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019. 132

[659] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 168, 197

[660] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 206

[661] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 206

[662] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018. 206

[663] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[664] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 206

[665] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023. 207

[666] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022. doi: 10.1145/3530811. 209