Information Processing and Management 62 (2025) 104196

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Collaborative local–global context modeling for session-based recommendation

Weiyue Li [a,1], Bowei Chen [b,1], Ming Gao [a,1], Jingmin An [a,1], Hao Dong [a], Cheng Chen [a], Weiguo Fan [c], Zhiguo Zhu [a,*]

[a] School of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, China
[b] Adam Smith Business School, University of Glasgow, United Kingdom
[c] Department of Business Analytics, Tippie College of Business, University of Iowa, United States

ARTICLE INFO

Keywords:
Session-based recommendation
Collaborative effect
Graph neural network
Disentangled attention

ABSTRACT

Session-based recommendation systems (SBRSs) predict the next item in a session by analyzing user interactions. While current methods emphasize sequential item relationships, they often overlook temporal information that highlights subtle shifts in user preferences. This gap can limit their ability to adapt to dynamic user behavior, and recent advances have yet to effectively integrate both sequential and non-sequential item transitions, which may lead to biased modeling. To address these limitations, this paper introduces Coase, a novel SBRS model that unifies local and global context modeling to capture fine-grained dynamic user preferences. Coase transforms session sequences into session star graphs, employing a Bi-Gated Graph Self-Attention Network for local context modeling, and introduces SudokuFormer to model time-aware sequential transitions within a global session context through disentangled attention and stable feature fusion. A triple attention mechanism is then utilized to fully integrate local and global contextual features. Comprehensive experiments conducted on four publicly available datasets demonstrate that Coase improves Recall by 1.71%–1.83%, Mean Reciprocal Rank (MRR) by 2.73%–2.80%, and Normalized Discounted Cumulative Gain (NDCG) by 2.32%–2.43% across the top 5, 10, 15, and 20 items. Ablation studies validate the framework and components of Coase, while additional analyses examine the effect of session length, and visualization studies illustrate diverse attention patterns. This research contributes a novel approach to SBRS, offering promising advancements in recommendation accuracy and user experience.

1. Introduction

Recommender systems play a crucial role in modern information filtering and decision-making by helping to alleviate information overload and facilitating efficient navigation on online platforms (Fang et al., 2020; Wu et al., 2023). Their widespread adoption across various platforms underscores their significant business value, particularly in managing and marketing customer relationships (Karimi et al., 2018). For example, Amazon attributes 35% of its product sales to recommendations (Hosanagar et al., 2014), while YouTube reports that 60% of clicks on its home screen are driven by its recommendation system (Davidson et al., 2010).

* Corresponding author at: No. 217 JianShan St., Shahekou District, Dalian, PR China.
E-mail address: zhu_zg@dufe.edu.cn (Z. Zhu).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.ipm.2025.104196
Received 9 March 2025; Received in revised form 12 April 2025; Accepted 12 April 2025
Available online 16 May 2025

Furthermore, personalized recommendations on Netflix are estimated to generate over $1 billion in annual value (Gomez-Uribe & Hunt, 2016).

Significant efforts have been made to design recommender systems across various contexts to provide personalized services, such as in point-of-interest recommendation (Zeng et al., 2025) and travel recommendation (Chen et al., 2024). However, many of these systems heavily depend on abundant side information to achieve satisfactory performance (Chen et al., 2024; Wei et al., 2024; Zeng et al., 2025). Acquiring such diverse side information can be challenging, which limits the scalability of these recommender systems. Additionally, many e-commerce recommender systems, especially those run by small retailers, as well as most news and media websites, typically do not track user IDs over extended periods to gather long-term interaction histories (Hidasi et al., 2016). While cookies and browser fingerprinting can provide some level of user recognition, these technologies are often unreliable and raise privacy concerns (Strycharz et al., 2021), leading to a lack of historical data. Furthermore, even when historical data is available for certain visitors, user preferences can be quickly influenced by external factors, making this historical data potentially unrepresentative of the user's current interests (Landia et al., 2022). As a result, delivering accurate recommendations during short user sessions becomes particularly challenging when both side information and historical data are limited.

To date, many SBRSs have been developed to deliver personalized content based solely on user behaviors within an ongoing anonymous session (e.g., recently viewed or purchased items). These systems enhance user satisfaction and engagement by adapting to rapidly changing preferences (Wang, Cao, et al., 2022). While existing SBRSs typically focus on understanding user preferences through the analysis of item transition sequences, they often overlook the temporal features embedded in the user's session context (Hidasi et al., 2016; Shalaby et al., 2022; Zhang, Xu, Ma, et al., 2024). Time-aware contextual features, such as dwell time on each item, serve as strong indicators of user preferences and content relevance (Fang et al., 2020), capturing levels of engagement without requiring additional external data. These insights not only enrich the context of a session but also help alleviate scalability issues associated with reliance on external data. Moreover, the variability in temporal features can significantly impact the accuracy of user modeling due to shifts in preferences (Dang et al., 2023, 2024). Although recent efforts have attempted to integrate temporal features into these models (Li et al., 2020; Wang et al., 2022), the methods employed often involve invasive techniques that fail to adequately capture the nuanced semantic relationships between items.

Several recent studies have transformed raw session sequences into various session graphs to model the underlying relationships between items within session contexts, achieving promising performance improvements (Chen & Wong, 2020; Wu et al., 2019; Xu et al., 2019; Yu et al., 2020). In particular, these graph learning-based methods effectively capture union-level and skip-behavior patterns, as demonstrated in Tang and Wang (2018), while sequence learning-based SBRSs struggle to identify these patterns. However, most graph learning-based SBRSs fail to account for the temporal dynamics of user sessions, specifically the sequence of item interactions and how these interactions influence evolving user preferences. Although absolute position encoding has been employed to address this issue (Pan et al., 2020), it relies on ineffective bootstrapping methods to combine contextual features extracted from both local and global context modeling. This results in a critical shortcoming: essential interactions lose their influence in the final recommendation process. Furthermore, users often engage in short sessions with limited interaction (Wang et al., 2019), leading to data sparsity in SBRSs. To mitigate this issue, leveraging cross-session data has been shown to improve performance (Wang, Chen, et al., 2023; Yu et al., 2023). However, despite the performance gains, these memory-based SBRS approaches require substantial memory resources, making them impractical for large-scale deployment in real-world scenarios.

To address the aforementioned challenges, we define a novel taxonomy to categorize existing SBRS research from the perspectives of local and global context modeling. Guided by the path modeling framework (Hui et al., 2009), we introduce a collaborative local–global contextual feature learning model named Coase, which effectively captures dynamic user preferences from different perspectives to tackle the next item prediction task. For local context modeling, we construct a session graph that preserves sequential transitions between adjacent items while also considering non-sequential transitions. The central node in each session graph is crucial, as it gathers contextual features and facilitates message passing among item nodes. We then employ the Bi-gated Graph Self-Attention Network (Bi-Gated GSAN) to learn the representations of both item nodes and the central node. In the realm of global context modeling, we incorporate learnable position encoding and time interval encoding to capture temporal dynamics. We also introduce the SudokuFormer, which analyzes temporal item transition relationships to account for user preferences. This model computes context-sensitive, disentangled attention weights based on item content, position encoding, and time intervals. To enhance model stability and effectively leverage temporal information, we have developed a stable, non-invasive fusion method. Additionally, our triple attention mechanism is designed to identify session-level preferences by considering the collaborative impact of both local and global context learning. This includes a mutual attention mechanism for learning long-term preferences by assigning weights to items based on their representations, collaborative impact, and the most recent interactions. Moreover, we employ two holistic attention networks to adaptively learn comprehensive preferences for both local and global context learning paths, with each path's preference influenced by long-term preferences, short-term preferences, and the overall collaborative effect.

Our study introduces several significant advancements in SBRSs:

- We propose a novel collaborative framework guided by the path modeling framework (Hui et al., 2009) that integrates graph and sequence learning paradigms. This unified approach not only enhances the accuracy of item recommendations but also emphasizes the collaborative effects essential for capturing fine-grained session-level user preferences.
- We introduce the Bi-Gated GSAN for the graph learning on the session star graphs, which highlights the feature of the node itself to alleviate information overwhelming.
- We present SudokuFormer, a novel approach that effectively analyzes the intricate contextual relationships between items, their positions within the session sequence, and the timing of interactions. This leads to more precise and stable attention weights.

- The effectiveness of Coase is rigorously evaluated across four datasets against 25 popular baseline models, consistently demonstrating superior performance. Additionally, ablation studies confirm the effectiveness of its framework and individual components.

The rest of the paper is organized as follows: Section 2 reviews the related literature; Section 3 presents our research objective and formalizes the session-based recommendation problem; Section 4 introduces technical details of the proposed Coase; Section 5 introduces our experimental setup and presents the results analysis; Section 6 concludes the paper by highlighting the theoretical contributions and practical implications of our work, as well as outlining potential future directions for research.

2. Related work

Recommender systems have made substantial progress in mitigating information overload. However, many users lack accessible external data, such as profiles or past interactions. To address this, SBRSs have been developed to generate recommendations based solely on dynamic, in-session preferences. In this section, we provide a comprehensive review of related SBRS research and outline the key insights that have shaped the development of our proposed Coase model.

2.1. Sequential and temporal information in session-based recommendation

The sequential order of user behaviors is crucial for tracking the evolution of user preferences, and existing SBRSs have adopted various methods to capture these preferences. For example, RNN-based methods (Hidasi et al., 2016; Li et al., 2017) track user preferences through causal learning, while transformer-based methods (Qiu, Huang, Chen, & Yin, 2022; Yin et al., 2024) introduce various position encodings to enhance the context-aware capacity. Moreover, GNN-based methods (Pan, Cai, Chen, Chen, & Chen, 2022; Wan et al., 2024; Wu et al., 2019) transform session sequences into graphs to capture complex item dependencies beyond simple sequential patterns. Furthermore, Transformer-based methods (Kang & McAuley, 2018; Xie et al., 2022) rely on explicit position encoding to maintain the notion of order (Huang et al., 2020), such as the absolute learnable position encoding (Gehring et al., 2017). However, these position-aware methods generally assume that the time intervals between items are the same and focus mainly on item sequence and position, without considering the different amounts of time users spend on each item, which can signal varying interest levels.

Recently, several time-aware methods have been proposed to capture temporal information from time intervals (Dang et al., 2023, 2024; Guo, Zhang, et al., 2022; Li et al., 2020; Wang et al., 2022; Wang, Yan, et al., 2023). For example, sequence data is augmented by incorporating time intervals (Dang et al., 2023, 2024), while sessions are segmented into specific time slices to capture dynamic user preferences (Wang, Yan, et al., 2023). Additionally, sequence positions are dynamically adjusted based on timestamps, using sinusoidal transformations to represent both absolute order and relative temporal proximity among points of interest (Wang et al., 2022). Compared to position-aware methods that focus solely on the sequential order of items, time-aware methods further emphasize the time intervals between those items. These time intervals reinforce the understanding of local user preferences, providing deeper insights into how users interact with items over time. However, many of time-aware methods rely on invasive and unstable methods for fusing item and temporal features, which can overwhelm the information and reduce training stability. Specifically, item features often lose significance when mixed with auxiliary temporal features using methods like summation or attention. Furthermore, the large volume of feature inputs can destabilize training, leading to suboptimal performance.

Various methods have been proposed to efficiently incorporate positional and temporal information into transformer models (Dufter et al., 2022). In this research, we focus on the methods for manipulating attention matrices, which refine attention weight calculations and lead to notable performance enhancements. For example, Shaw et al. (2018) split the computation of original attention weights to prevent broadcasting relative position representations. Chen, Tsai, et al. (2021) found that adding position encoding at the input stage resulted in poorer performance and separated the impact of incorporating relative position features. Dai et al. (2019) re-parameterized the four terms decomposed by the vanilla attention score to enhance generalization. Wu et al. (2021) employed re-scaled coefficients to adjust the raw attention weights computed by the dot product of query and key. Dufter et al. (2020) eliminated word-position terms and substituted the position–position term with a learnable matrix. However, these methods either overlook the efficacy of multiple temporal information sources or fail to address the intricate relationship between different temporal information and input elements. Different from the above methods, DeBERTa (He et al., 2021) computed attention weights among input elements using disentangled matrices based on their content and relative positions. However, it overlooked the critical impact of absolute position on self-attention weights.

The above challenges motivates the design of our SudokuFormer. Specifically, recent marketing literature (Ursu et al., 2023) finds that consumers may take breaks from information acquisition because of fatigue. Similarly, another study (Li et al., 2024) claims that users may be tired of the recommendations that are too similar to content they have been exposed to in a short historical period. Therefore, we introduce learnable position encoding to track the subtle changes in user preferences over time. Moreover, the time cues, such as dwell time on each item and switching time between item-pairs, may reflect the user's degree of interest (Fang et al., 2020). For example, the more time a user spends on an item, the more likely they are to be satisfied with its contents (Kim et al., 2014; Zhang et al., 2014). Additionally, some works believe that items within a close time interval usually have a similar user interest involving a smaller interest drift (Tang et al., 2022; Wang, Zeng, et al., 2023). Therefore, we parameterize time cues as time interval encoding to capture the user's unique taste for items. Furthermore, we extend the original disentangled attention mechanism to account for complex relations among various inputs. Drawing inspiration from Press et al. (2021), Liu et al. (2021),

Table 1
Summary of key techniques of the related literature and our study.

Orientation	Models	Backbone	Position-aware	Time-aware	Sequence-based	Graph-based
Local	NextItNet (Yuan et al., 2019)	CNN			✔	
	Grec (Yuan et al., 2020)	CNN			✔	
	MIHSG (Guo, Yang, et al., 2022)	GNN				✔
	FineRec (Zhang, Xu, Wu, et al., 2024)	GNN				✔
	SRGNN (Wu et al., 2019)	GNN + Attention				✔
	CGL (Pan, Cai, Chen, Chen, & Chen, 2022)	GNN + Attention				✔
	GCARM (Pan, Cai, Chen, & Chen, 2022)	GNN + Attention				✔
	TASER (Ye et al., 2020)	GNN + Attention		✔		✔
Global	GRU4Rec (Hidasi et al., 2016)	RNN			✔	
	NARM (Li et al., 2017)	RNN + Attention			✔	
	TiSASRec (Li et al., 2020)	Attention	✔	✔	✔	
	STiSAN (Wang et al., 2022)	Attention	✔	✔	✔	
	RETR (Yao et al., 2024)	Attention	✔		✔	
	MEGAN (Wang, Zhang, et al., 2024)	GNN + Attention				✔
	HIDE (Li et al., 2022)	GNN + Attention	✔			✔
Dual	TARN (Zhang, Cao, et al., 2022)	CNN	✔	✔	✔	
	CM-GNN (Wang, Gao, et al., 2023)	GNN	✔			✔
	AdaMCT (Jiang et al., 2023)	CNN + Attention	✔		✔	
	H3GNN (Yin et al., 2024)	GNN + Attention	✔			✔
	PTGCN (Huang et al., 2023)	GNN + Attention	✔	✔		✔
	EMBSR (Yuan et al., 2022)	GNN + RNN + Attention	✔		✔	✔
	RESTC (Wan et al., 2024)	GNN + Attention	✔		✔	✔
	GES-SASRec (Zhu et al., 2023)	GNN + Attention	✔		✔	✔
	Coase (Our study)	GNN + Attention	✔	✔	✔	✔

we combine item encoding with position encoding and time interval encoding as keys and queries, while retaining item encoding as values. This method effectively leverage the rich temporal and positional information without increasing parameters or runtime. After that, we propose the stable non-invasive feature fusion method to integrate various temporal features without compromising the consistency of the item feature space, while the additional normalization enhances training stability.

2.2. Revisiting sequence-based and graph-based SBRSs from a context modeling view

Existing SBRSs have shown promising results but the factors contributing to their performance improvements remain poorly understood as current taxonomies based on neural network backbones fail to provide a comprehensive view of the elements influencing SBRS performance. Depending on the data structure, SBRSs can be categorized into sequence-based and graph-based approaches. Sequence-based SBRSs typically use fixed-length session representations, where longer sessions are truncated and shorter ones are padded. These methods are divided into three main groups based on the backbone network: CNN-based, RNN-based, and transformer-based. CNN-based methods focus on extracting local context using convolutional layers to hierarchically aggregate patterns into a global understanding of user preferences (Tuan & Phuong, 2017; Yuan et al., 2020). RNN-based methods capture long-term user preferences by learning sequential dependencies throughout the session (Hidasi et al., 2016; Li et al., 2017), while transformer-based methods model global relationships via self-attention and position encoding (Li et al., 2020; Sun et al., 2019). Graph-based SBRSs, on the other hand, transform session sequences into session graphs to capture item relationships. These methods include vanilla item transition graphs, heterogeneous item transition graphs, and hyper-graphs. Vanilla methods focus on sequential item transitions within sessions , while heterogeneous graphs add session nodes for global contextual learning (Pan et al., 2020; Yuan et al., 2022). Hyper-graphs go beyond pairwise item relationships to capture complex interactions by aggregating features from hyper-nodes to hyper-edges (Xia et al., 2021; Yin et al., 2024).

As summarized in Table 1, we offer a new perspective to explore how context modeling orientation affects the effectiveness of SBRSs. Here the term context captures all session information used for recommendations (Wang, Cao, et al., 2022), such as item contents and their sequential order. We classify existing work into three categories based on the contexts that different models primarily focus on, including local-oriented methods, global-oriented methods, and dual-oriented methods. The local context refers to item transitions contained within a session fragment or the neighborhood of a session graph, highlighting the immediate interactions and dependencies between items that are closely related in time or interaction sequence. By contrast, the global context encompasses item transitions across the entire session, highlighting long-range dependencies and overall patterns that span beyond immediate interactions.

Both local-oriented and global-oriented SBRSs aim to discern user preferences, offering distinct perspectives. Local-oriented methods excel at extracting detailed user intents from session segments and aggregating them to form a coherent understanding of preferences within the current session. However, while they allow for fine-grained analysis, they can suffer from over-reliance on local context, potentially missing long-term consistency and stable preferences. In contrast, global-oriented methods focus on comprehensive user preferences to guide interactions, adapting to interest drift and prioritizing core needs. However, they often struggle to capture the nuances of local context and rapid preference changes. Thus, integrating both approaches offers a more

complete and adaptive understanding of user behavior. CM-GNN (Wang, Gao, et al., 2023) and H3GNN (Yin et al., 2024) are representative multi-graph methods that learn local item features from vanilla session graphs and global item transitions, refining global features using learnable position encoding and hyper-graph set relations. AdaMCT (Jiang et al., 2023), a sequence-based method, incorporates locality bias into Transformers by combining global attention with local convolutional filters. GES-SASRec (Zhu et al., 2023) integrates local and global context modeling by considering semantic item relations. However, the bootstrapping frameworks used by CM-GNN, H3GNN, and GES-SASRec struggle to distinguish independent item relations, while AdaMCT faces challenges in separating sequential and non-sequential dependencies due to feature mixture at each layer. Overall, while SBRSs have progressed in capturing local and global contexts, challenges remain in effectively integrating contextual features and balancing short-term adaptability with long-term consistency.

2.3. Insights for the development of coase

Our review identifies key limitations in existing SBRSs, which often fail to differentiate between local and global item dependencies. To address these challenges, we propose the Coase model, drawing on the path modeling framework (Hui et al., 2009). This framework emphasizes the value of user path data, which in this case refers to non-physical, discrete interactions of anonymous users on online platforms. Based on this, Coase adopts a dual-path approach to capture both local and global contextual features. The Bi-Gated GSAN module is employed for local context modeling, capturing non-sequential item dependencies within session star graphs, while SudokuFormer handles global context by modeling sequential item transitions across sessions. To align these perspectives, Coase integrates a triple attention mechanism that harmonizes user preferences across local and global paths, considering a wide range of contextual features and their collaborative effects. The path modeling framework also highlights that user paths alone may not reliably indicate their true intentions, as users with different goals follow distinct paths (Hui et al., 2009). To tackle this issue, Coase employs an end-to-end method that continuously tracks dynamic user preferences by updating user and item representations in real-time based on their interactions. This approach avoids the potential biases of self-reported data (de Reuver & Bouwman, 2015), allowing the model to adapt to users' evolving interests more effectively. Finally, the path modeling framework underscores the importance of accounting for user heterogeneity to understand fine-grained preferences (Larsen et al., 2020). Coase addresses this by analyzing several key factors, such as user–item interactions, time cues, and both sequential and non-sequential behavior patterns. Notably, while social effects are considered minimal in some cases (Hui et al., 2009), Coase focuses solely on user–item interaction data due to the anonymity of users, which aligns with prior research that excludes social cues when studying anonymous behavior patterns (Chen, Burke, Hui, & Leykin, 2021; Fisher & Woolley, 2024; Larsen et al., 2020). By focusing on these key elements, Coase provides a robust framework for capturing both immediate and evolving user preferences in session-based recommender systems.

3. Research objective

The objective of this study is to capture user preferences within a time-sensitive session context to recommend the next item aligned with the user's evolving, fine-grained intentions. To achieve this, we outline a general mathematical framework. The model first identifies all unique items, represented as the set V with size $|V|$, and tracks an anonymous user's current session as a sequence of interacted items s with size $|s|$. Given this session history, our model aims to predict the next item $v^s_{|s|+1}$ that the user will interact with. To achieve this, our model first creates a numerical representation (embedding) for each item $e_i \in \mathbb{R}^d$, where d represents the embedding dimension, and refines these embeddings according to session context and item relationships. The model further incorporates mutual preference, short-term preference, and long-term preference to refine its understanding of user preferences. Ultimately, it recommends items by calculating a score that integrates the user's comprehensive preferences and candidate item representations, presenting the top items with the highest scores as recommendations. Table 2 summarizes the fundamental notations, while the subsequently derived notations are omitted.

4. Coase

Fig. 1 illustrates the architecture of the proposed Coase, which includes several key components. The embedding module generates initial representations for items, their session positions, and the time intervals between interactions. The local contextual feature extractor constructs a session-specific graph and employs a Bi-Gated GSAN to capture item relationships and their significance. The global contextual feature extractor models the order of interactions by incorporating learnable position and time interval encodings, allowing for a better understanding of how sequence affects user preferences. The SudokuFormer refines item representations using a disentangled attention mechanism and a stable, non-invasive feature fusion approach, enhancing the understanding of each item's role. The session-level co-learning module integrates local and global context modeling through a triple attention mechanism, capturing comprehensive user preferences by considering short-term, contextual, and long-term preferences. Finally, the prediction layer generates scores for candidate items and produces the top items recommendation list. The learning process of Coase is summarized in Algorithm 1.

Table 2
Summary of key notations.

Notation	Description
V, S	Item set, session sequence set
$e_{(\cdot)}$, $E_{(\cdot)}$	Item embedding vector, item embedding matrix
d	Embedding dimension
G_s	Session star graph of session s
$\|\mathcal{E}\|$	The edge number in the session star graph
V_s	Unique item node set of session s
$\alpha_{(\cdot)}$, $\beta_{(\cdot)}$, $\gamma_{(\cdot)}$, $\delta_{(\cdot)}$, $\mu_{(\cdot)}$, $\varphi_{(\cdot)}$, $\xi_{(\cdot)}$	Attention weight
σ	Sigmoid activation function
$[\cdot\|\|\cdot]$	Concat operation
$W_{(\cdot)}^{(\cdot)}$	Learnable parameter
$head_{(\cdot)}^{(\cdot)}$	Attention head
N	Attention head number
$center_{(\cdot)}^{(\cdot)}$	central node feature
$L_{(\cdot)}$	Stacking layer number
P_A^Q, P_A^K	Learnable position embedding matrices for queries and keys
P_R^Q, P_R^K	Learnable time interval embedding matrices for queries and keys
$t_{(\cdot)}$	Timestamp of interaction with the item
$span_{(\cdot)}$	Time interval
m	The predefined maximum session length
u	The number of unique item within the session
$LN(\cdot)$	Layer normalization
$MLP(\cdot)$	Multi-Layer perceptron
$h_{short}^{(\cdot)}$, $h_{context}^{(\cdot)}$, $h_{long}^{(\cdot)}$	Representations of short-term preference, contextual preference, and long-term preference
$h_{com}^{(\cdot)}$	Comprehensive preference
τ	Temperature hyper-parameter
$y_{(\cdot)}$	The recommendation score for the item
λ	The hyper-parameter to balance the weights of different losses

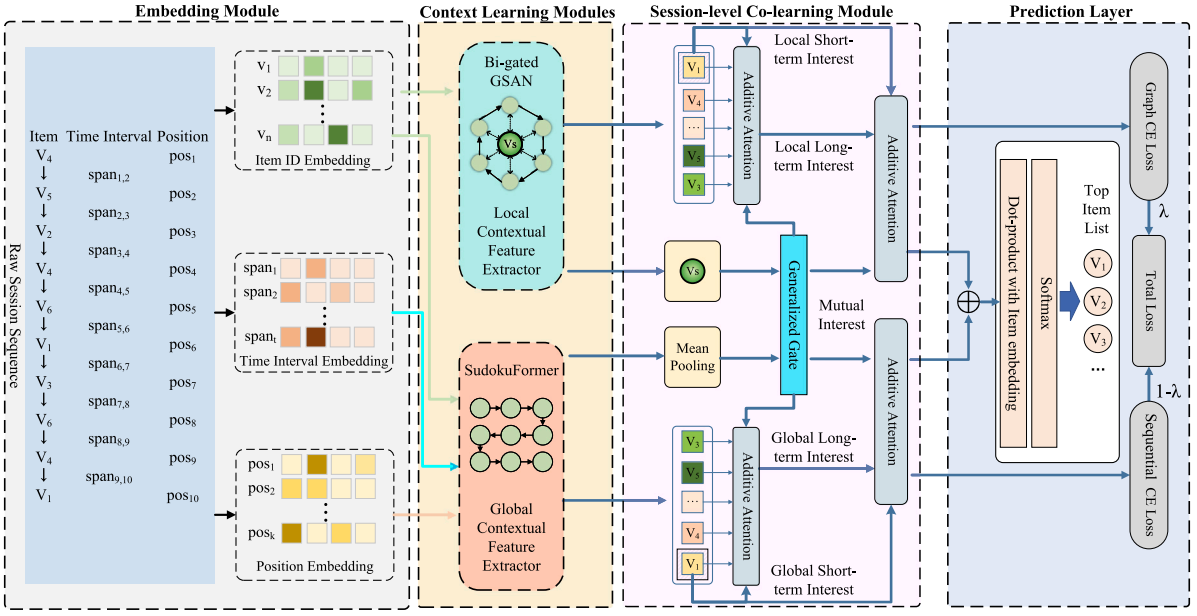


Fig. 1. Schematic view of the proposed Coase, featuring key modules: the embedding module for generating item, session, and time interval representations; the context learning modules including local and global contextual feature extractors to capture non-sequential dependencies and sequential transitions among items; the session-level co-learning module, which integrates short- and long-term user interests through a triple attention mechanism; and the prediction layer that ranks items and outputs the top recommendations.

4.1. Local contextual feature extractor

As shown in Fig. 2, the local contextual feature extractor is designed to learn item representations within the session star graph using the Bi-Gated GSAN. This module consists of two main components: session star graph construction and the Bi-Gated GSAN.

Algorithm 1 The forward propagation flow of Coase

Input: The session sequences S.
Output: Top-k recommendation items at the next time step.

/ * ***Embedding module*** * /
Initialize the item embedding matrix E, two time interval embedding matrices P_R^Q and P_R^K, and two position embedding matrices P_A^Q and P_A^K;
/ * ***Context learning modules*** * /
/ ** *Local contextual feature extractor* ** /
Construct session star graph G_s;
Initialize the feature of the central node v_s: $center_s^g = \frac{1}{u}\sum_{v_i \in s} e_i$;
for each layer of Bi-Gated GSAN **do**
  for each session sequence $s \in S$ **do**
    Update the features of each item node based on the neighbor item node features with $Eq.$ (1) to $Eq.$ (5);
    Update the features of each item node based on the central node features with $Eq.$ (6) and $Eq.$ (7);
    Update the central node features based on item node features with $Eq.$ (8) and $Eq.$ (9);
  end for
end for
Summarize all the Bi-Gated GSAN layers to get the local context-aware item features with $Eq.$ (10) and $Eq.$ (11);
/ ** *Global contextual feature extractor* ** /
for each layer of SudokuFormer **do**
  for each session sequence $s \in S$ **do**
    Get the three query matrices, the three key matrices, and the value matrix with $Eq.$ (14) to $Eq.$ (16);
    Calculate the multi-head disentangled attention weights with $Eq.$ (17) to $Eq.$ (19);
    Update the item features with $Eq.$ (20);
  end for
end for
Summarize all the SudokuFormer layers to get the global context-aware item features;
/ * ***Session-level co-learning module*** * /
for each session sequence $s \in S$ **do**
  Get the short-term preferences: $h_{short}^g = e_{last}^g$ and $h_{short}^{se} = e_{last}^{se}$;
  Extract the contextual preferences: $h_{context}^g = center_s^g$ and $h_{context}^{se} = \frac{1}{|S|}\sum_{v_i \in S} e_i^{se}$;
  Calculate the long-term preferences h_{long}^g and h_{long}^{se} with $Eq.$ (22) to $Eq.$ (27);
  Fuse the various preferences by holistic attention mechanism with $Eq.$ (28) to $Eq.$ (30);
end for
/ * ***Prediction layer*** * /
for each candidate item $v_i \in V$ **do**
  Calculate the interaction probability: $y_i \leftarrow Eq.$ (31);
end for
Get predicted interaction probability list: $[\hat{y}_1, \hat{y}_2, \dots]$;
Select the items with top-K predicted interaction probabilities to form the recommendation list.

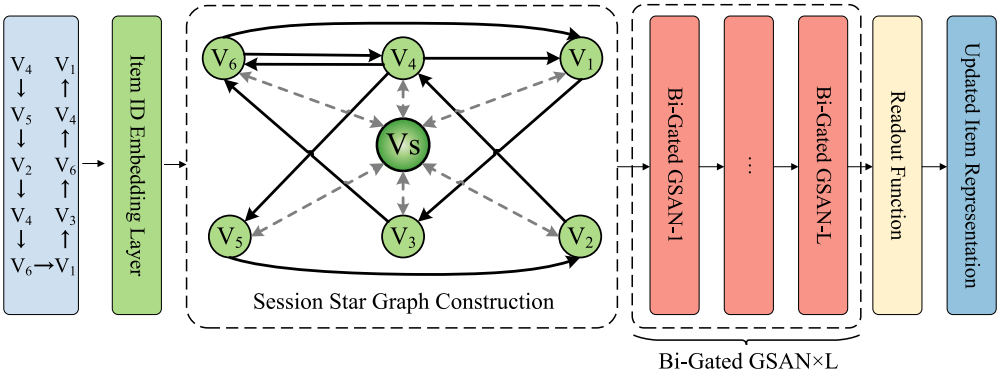


Fig. 2. Schematic view of the local contextual feature extractor, in which the session star graph is constructed based on the raw session sequence and the additional session center node; the stacking Bi-Gated GSANs are adopted to learn the representations of item nodes; and the readout function is introduced to aggregate the features learned by each Bi-Gated GSAN layer.

The session star graph organizes items into a structured graph, while the Bi-Gated GSAN processes this graph to capture and refine item relationships, enhancing each item's representation in the session context.

4.1.1. Session star graph construction

As shown in the upper left part of Fig. 1, given the current session sequence $s = \{v_4, v_5, v_2, v_4, v_6, v_1, v_3, v_6, v_4, v_1\}$, the unique item node set $V_s = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ is derived from the session sequence, while directed edges are created based on the pairwise transition relationships between adjacent items. Furthermore, the central node v_s is introduced to enhance the connectivity of the vanilla session graph (Guo et al., 2019). It serves as an intermediary node, facilitating the propagation of information from items that do not have a direct connection in a two-hop manner. Specifically, bidirectional edges are added between the central node and each item node in the session star graph G_s to improve connectivity (Pan et al., 2020).

4.1.2. Graph-based item representation learning

After constructing the session star graph, the bi-gated graph self-attention network is adopted to update the representation of the item nodes and the central node. Firstly, the item node representation is initialized based on the item embedding, while the dropout layer (Srivastava et al., 2014) is performed to alleviate the over-fitting and improve the robustness (Du, Yuan, Zhao, Fang, et al., 2023; Du, Yuan, Zhao, Qu, et al., 2023). Subsequently, the central node representation is initialized by the average pooling on the item nodes. After that, the self-attention mechanism is adopted to compute the influence of the neighbor item nodes as follows:

$$\alpha_{i,j} = softmax\left(\frac{\left(W_1^g e_i\right)^T \left(W_2^g e_j\right)}{\sqrt{d}}\right), \tag{1}$$

where $\alpha_{i,j}$ denotes the attention weight of the target item node v_i and one of its neighbor item nodes v_j, W_1^g and $W_2^g \in \mathbb{R}^{d\times d}$ are the trainable matrices. The bias term is omitted for briefly.

Then the first gating network is used to update the representation of the target item node v_i as follows:

$$e_i' = \beta_i W_3^g e_i + \left(1-\beta_i\right) neigh_i, \tag{2}$$

$$\beta_i = \sigma\left(W_\beta^T \left[W_3^g e_i \left\| neigh_i \right\| W_3^g e_i - neigh_i\right]\right), \tag{3}$$

$$neigh_i = W_4^g \sum_{j\in N(i)} \alpha_{i,j} e_j, \tag{4}$$

where σ denotes the sigmoid activation function, e_i' denotes the updated representation of the target item node v_i based on the item transition relationships, $\left[\cdot\right\| \cdot \left\|\cdot\right]$ denotes the concat operation among three tensors, W_3^g, $W_4^g \in \mathbb{R}^{d\times d}$, and $W_\beta \in \mathbb{R}^{3d}$ are the trainable parameters.

Subsequently, the multi-head mechanism is employed to capture the multi-aspect transition relationships among item nodes. For brevity, we denote the above process as $e_i' = head^g\left(e_i, E_{neigh}\right)$, where E_{neigh} is the embedding matrix of the neighbor item nodes. The multi-head mechanism is defined as follows:

$$\hat{e}_i = \sum_{n=1}^{N} head_n^g W_5^g, \tag{5}$$

where n denotes the number of attention heads, $W_5^g \in \mathbb{R}^{d\times d}$ denotes the trainable parameter.

The representation of the target node is further updated based on the central node feature by the second gating network as follows:

$$\tilde{e}_i^g = \gamma_i W_6^g \hat{e}_i + \left(1-\gamma_i\right) W_7^g center_s, \tag{6}$$

$$\gamma_i = \sigma\left(W_\gamma^T \left[W_6^g \hat{e}_i \left\| W_7^g center_s \right\| W_6^g \hat{e}_i \odot W_7^g center_s\right]\right), \tag{7}$$

where $\tilde{e}_i^g$ denotes the local contextual representation of the target item node, $center_s \in \mathbb{R}^d$ denotes the central node feature of the session s, $\odot$ denotes the Hadamard product, W_6^g, $W_7^g \in \mathbb{R}^{d\times d}$, and $W_\gamma \in \mathbb{R}^{3d}$ are the trainable parameters.

Next, the central node feature can be updated:

$$center_s^g = \frac{1}{u}\sum_{v_i\in s} \delta_{i,s} \tilde{e}_i^g, \tag{8}$$

$$\delta_{i,s} = softmax\left(center_s^T \tilde{e}_i^g\right), \tag{9}$$

where $center_s^g$ denotes the updated central node representation, u denotes the number of unique item within the session. For brevity, we denote the above process as $\tilde{e}_i^g = GSAN\left(e_i, E_{neigh}, center_s\right)$. Then, the bi-gated graph self-attention network can be stacked to capture the distant item transitions. Finally, we adopt the simple but effective readout function (He et al., 2020) to learn the final item node feature:

$$\tilde{e}_i^{g,(l)} = GSAN\left(e_i^{(l-1)}, E_{neigh}^{(l-1)}, center_s^{(l-1)}\right), \tag{10}$$

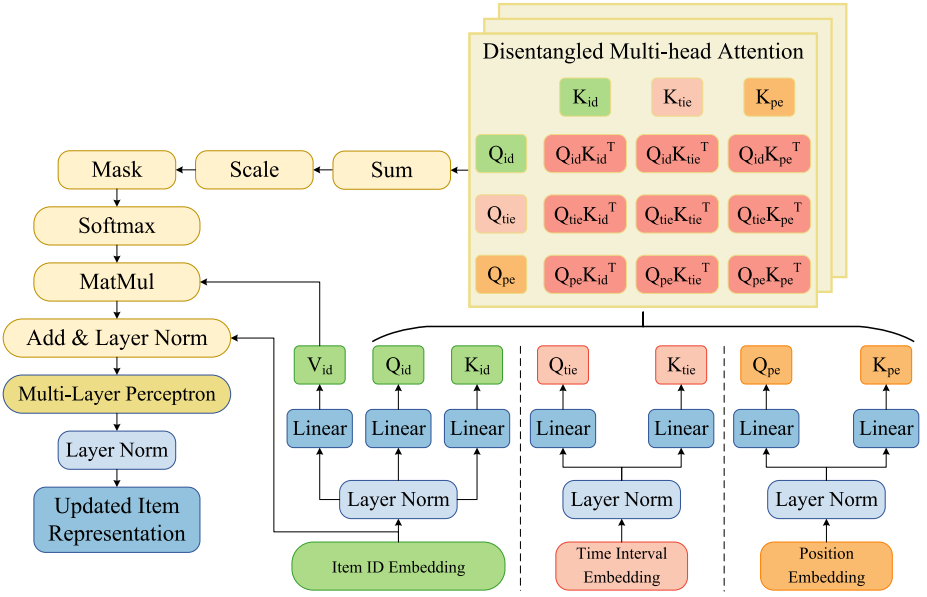


Fig. 3. Schematic view of SudokuFormer. Instead of mixing various encodings as input, the proposed SudokuFormer captures the fine-grained correlation between any two of the item content, item position, and the time interval based on the disentangled attention weights. Moreover, the dual layer norm method and the non-invasive fusion method is introduced to improve the quality of the item representations.

$$e_i^g = \frac{1}{L_{gnn}} \sum_{l=1}^{L_{gnn}} e_i^{\sim g,(l)}, \tag{11}$$

where $e_i^{g,(l)}$ denotes the target item node representation involved in l-order local contextual information.

4.2. Global contextual feature extractor

The global contextual feature extractor is designed to update item representations by incorporating positional and temporal information. Specifically, it assumes that interactions with an item are influenced by the item's content, position in the session, and the time cues. These factors are parameterized as learnable item encoding, position encoding, and time interval encoding, respectively. Following this, the proposed SudokuFormer works to disentangle the hidden relationships among these factors and updates the item representation using a stable, non-invasive fusion method, as illustrated in Fig. 3.

4.2.1. Input encodings provided by embedding module

Let $E_{se} \in \mathbb{R}^{|S| \times d}$ denotes the embedding matrix of the items involved in the session. The dropout layer (Srivastava et al., 2014) is performed to stabilize the training process (Du, Yuan, Zhao, Fang, et al., 2023; Du, Yuan, Zhao, Qu, et al., 2023). Furthermore, the learnable position embedding is performed to consider the positional information, which accounts for how the user's progress through the session impacts their interaction with the item. Inspired by Li et al. (2020) and Shaw et al. (2018), two distinct learnable position embedding matrices $P_A^Q \in \mathbb{R}^{m \times d}$ and $P_A^K \in \mathbb{R}^{m \times d}$ are adopted for queries and keys in the disentangled attention mechanism of the proposed SudokuFormer, respectively.

$$P_A^Q = \left\{ p_1^{a,q}, \ \ldots, \ p_m^{a,q} \right\}, \qquad P_A^K = \left\{ p_1^{a,k}, \ \ldots, \ p_m^{a,k} \right\}, \tag{12}$$

where m denotes the predefined maximum session length. Conceptually, the positional encoding gives the model a temporal clue or "bias" about how information should be gathered, i.e., where to attend (Dai et al., 2019). Moreover, the learnable time interval embedding captures the levels of user engagement on the item by mapping time cues onto a low-dimensional space. Formally, we model the time interval $span_{i,j} = \left| t_j - t_i \right| \ (j > i)$ as the representation of the time cues on the item v_i, where t_i and t_j denotes the timestamp of the item v_i and the next interacted item v_j, respectively.

Although a longer dwell time increases the likelihood of visiting detailed modules such as reading comments and specifications (Gong & Zhu, 2022), precise dwell time is not useful beyond a certain threshold (Li et al., 2020). Because excessive dwell time implies some abnormal behavior such as prolonged inactivity and background usage. Moreover, considering excessive dwell time also complicates calculations and introduce unnecessary parameters, which improves the risk of over-fitting. Therefore, the maximum time interval between two items is clipped to the specified threshold (Li et al., 2020; Shaw et al., 2018), denoted as z. Formally, the clip operation $span_{i,j}^{clip} = min\left(z, \left| t_j - t_i \right| \right)$ is applied to each time interval.

Next, two distinct learnable time interval embedding matrices $P_R^Q \in \mathbb{R}^{m\times m\times d}$ and $P_R^K \in \mathbb{R}^{m\times m\times d}$ are similarly adopted for queries and keys in the disentangled attention mechanism of the proposed SudokuFormer, respectively.

$$P_R^Q = \begin{bmatrix} p_{1,1}^{r,q} & \cdots & p_{1,m}^{r,q} \\ \cdots & \cdots & \cdots \\ p_{m,1}^{r,q} & \cdots & p_{m,m}^{r,q} \end{bmatrix}, \qquad P_R^K = \begin{bmatrix} p_{1,1}^{r,k} & \cdots & p_{1,m}^{r,k} \\ \cdots & \cdots & \cdots \\ p_{m,1}^{r,k} & \cdots & p_{m,m}^{r,k} \end{bmatrix}. \tag{13}$$

Subsequently, the embedding matrix of the items E_s, the learnable position and time interval embedding matrices for queries and keys, i.e., P_A^Q, P_A^K, P_R^Q, and P_R^K, are adopted as the initial input of proposed SudokuFormer.

4.2.2. Sequence-based item representation learning

Our SudokuFormer is adopted to capture the time-aware sequential item transition patterns based on the item encoding, the position encoding, and the time interval encoding. Specifically, the disentangled attention mechanism (Wang, Ma, et al., 2024, 2023) is introduced to understand the mutual effect among item content, position, and time interval. Moreover, the dual layer norm is employed to improve the stability of the attention mechanism (Wang, Ma, et al., 2024, 2023). Furthermore, the non-invasive fusion method (Liu et al., 2021) is adopted to avoid temporal information overwhelms item representation, which maintains the consistency of embedding space and updates the item representation more efficiently. Formally,

$$Q_C = LN\left(E_{se}\right) W_1^{se}, \qquad K_C = LN\left(E_{se}\right) W_2^{se}, \qquad V_C = LN\left(E_{se}\right) W_3^{se}, \tag{14}$$

$$Q_A = LN\left(P_A^Q\right) W_4^{se}, \qquad K_A = LN\left(P_A^K\right) W_5^{se}, \tag{15}$$

$$Q_R = LN\left(P_R^Q\right) W_6^{se}, \qquad K_R = LN\left(P_R^K\right) W_7^{se}, \tag{16}$$

$$\hat{E}_{se} = multihead\left(Q_C, K_C, Q_A, K_A, Q_R, K_R, V_C\right) = dropout\left(LN\left(concat\left(head_1, \ldots, head_f\right)\right)\right) W_8^s, \tag{17}$$

$$head_i = softmax\left(\frac{A}{\sqrt{d}}\right) V_C, \tag{18}$$

$$A = \left(Q_C, Q_A, Q_R\right)^T \left(K_C^T, K_A^T, K_R^T\right), \tag{19}$$

where $LN(\cdot)$ denotes the layer normalization, $dropout(\cdot)$ denotes the dropout (Srivastava et al., 2014), $multihead(\cdot)$ denotes the multi-head mechanism and the computation of head i is denoted as $head_i$. The scale factor $\sqrt{d}$ is used to avoid large values of the inner product. W_1^{se}, W_2^{se}, W_3^{se}, W_4^{se}, W_5^{se}, W_6^{se}, $W_7^{se} \in \mathbb{R}^{d\times d_{head}}$, and $W_8^{se} \in \mathbb{R}^{d\times d}$ denote trainable parameters.

Inspired by Dai et al. (2019), each term has its intuitive meaning under the new parameterization: (i) The content-to-content term $Q_C K_C^T$ represents content-based addressing; (ii) The content-to-position term $Q_C K_A^T$ captures a position-dependent content bias; (iii) The content-to-time interval term $Q_C K_R^T$ captures a time interval-dependent content bias; (iv) The position-to-content term $Q_A K_C^T$ captures an content-dependent positional bias; (v) The position-to-position term $Q_A K_A^T$ represents position-based addressing; (vi) The position-to-time interval term $Q_A K_R^T$ captures an time interval-dependent positional bias; (vii) The time interval-to-content term $Q_R K_C^T$ captures a content-dependent time interval bias; (viii) The time interval-to-position term $Q_R K_A^T$ captures a position-dependent time interval bias; (ix) The time interval-to-time interval term $Q_R K_R^T$ represents time interval-based addressing.

In comparison, the formulation in Shaw et al. (2018) has terms (i) and (iii); the formulation in Chen, Tsai, et al. (2021) has terms (i) and (v); the formulation in Yuan et al. (2022) has terms (i), (iii), and (vii); the formulation in Li et al. (2020) has terms (i), (ii), and (iii); the formulation in Dai et al. (2019) has terms (i), (iii), (vii), and (ix); the formulation in Vaswani et al. (2017) has terms (i), (ii), (iv), and (v). It can be assumed that the superior performance of self-attention comes from multiplicative interaction which provides the powerful inductive bias (Jayakumar et al., 2020). We argue that all the terms are significant since the attention weight of item-pairs depend on the involvement of the item content, the positional information, and the time interval. Therefore, we keep all the interaction terms among the three types of encoding.

Also, Chen, Tsai, et al. (2021), Shaw et al. (2018), and Li et al. (2020) merge the linear projection $P_R^Q W$ into a single trainable matrix $\hat{P}_R^Q$, which eliminated the helpful inductive bias (Dai et al., 2019). However, retaining the inductive bias facilitates the learning of robust and transferable patterns during model training, which improves resilience when dealing with novel items and noisy data during evaluation. Therefore, we keep all linear projections when computing the query, key, and value.

Similar to the traditional self-attention mechanism, the disentangled attention mechanism in our SudokuFormer remains a linear model, lacking the ability to capture non-linear relationships. Therefore, the feed-forward network is considered after the self-attention mechanism, which comprises a Multi-Layer Perceptron (MLP) with GELU activation and the dual layer norm method (Wang, Ma, et al., 2024, 2023) as follows:

$$\tilde{E}_{se} = MLP\left(\hat{E}_{se}\right) = dropout\left(LN\left(GELU\left(LN\left(\hat{E}_{se}\right) W_9^{se}\right)\right)\right) W_{10}^{se}, \tag{20}$$

where $W_9^{se} \in \mathbb{R}^{d\times 4d}$ and $W_{10}^{se} \in \mathbb{R}^{4d\times d}$ denote the trainable parameters. Subsequently, we stack the attention blocks and adopt the similar readout function in Eq. (11) to capture high-order sequential item transition relations. The updated item representation is defined as E_{se}.

4.3. Session-level co-learning module

After respectively updating the item representations in local and global context modeling paths, the session-level co-learning module is adopted to learn the comprehensive preferences. Specifically, for each path, the short-term preference is denoted as the last item. Moreover, the central node feature and the average of the item features within the current session are viewed as the representations of the contextual preferences for local and global context modeling, respectively. Furthermore, the triple attention mechanism is adopted to learn the comprehensive preference of each path. Specifically, the mutual attention mechanism is adopted to learn the long-term preference with the consideration of the short-term preference and the collaborative effect. After that, the holistic attention mechanism learns the comprehensive preference by integrating collaborative effect, short-term preference, and long-term preference.

Following Pan, Cai, Chen, Chen, and Chen (2022) and Pan, Cai, Chen, and Chen (2022), we consider the representation of the last item to reflect the short-term preference, i.e., $h^g_{short} = e^g_{last}$ and $h^{se}_{short} = e^{se}_{last}$, for graph and sequential learning, respectively. Furthermore, the center node representation is considered as the local contextual preference, i.e., $h^g_{context} = center^g_s$, while the global contextual preference $h^{se}_{context}$ is learned based on averaging the representation of the interacted items as follows:

$$h^{se}_{context} = \frac{1}{|S|}\sum_{v_i \in S} e^{se}_i. \tag{21}$$

The contextual preferences offer a stable perspective for representing a user's general interests, mitigating the effects of accidental behaviors and interest drift. To capture this, we extract the representation of the collaborative effect between the local and global context modeling paths, based on these contextual preferences. Following this, a mutual attention mechanism is applied to learn long-term preferences while accounting for the collaborative effect:

$$h^g_{long} = \sum_{v_i \in S} \mu_i e^g_i, \tag{22}$$

$$\mu_i = softmax\left(W^T_\mu \sigma\left(\left[W^{long}_1 e^g_i \left\| W^{long}_2 h_{mul}\right\| W^{long}_3 h^g_{short}\right]\right)\right), \tag{23}$$

$$h^{se}_{long} = \sum_{v_i \in S} \varphi_i e^{se}_i, \tag{24}$$

$$\varphi_i = softmax\left(W^T_\varphi \sigma\left(\left[W^{long}_4 e^{se}_i \left\| W^{long}_5 h_{mul}\right\| W^{long}_6 h^{se}_{short}\right]\right)\right), \tag{25}$$

$$h_{mul} = \eta h^g_{context} + (1-\eta) h^{se}_{context}, \tag{26}$$

$$\eta = \sigma\left(W^T_\eta \left[h^g_{context} \| h^{se}_{context} \| h^g_{context} \odot h^{se}_{context}\right]\right), \tag{27}$$

where $h_{mul} \in \mathbb{R}^d$ denotes the mutual preference representation. $h^g_{long} \in \mathbb{R}^d$ and $h^{se}_{long} \in \mathbb{R}^d$ denote the long-term preferences for graph and sequential learning, respectively. W^{long}_1, W^{long}_2, W^{long}_3, W^{long}_4, W^{long}_5, and $W^{long}_6 \in \mathbb{R}^{d\times d}$ denote the trainable parameters. W^T_μ and $W^T_\varphi \in \mathbb{R}^{3d}$ denote the trainable vectors. The mutual attention mechanism accentuates complementary information to improve the performance, while mitigates the impact of redundant information from local and global context modeling paths.

The comprehensive preference for each path is then learned by adaptive aggregating the short-term, long-term, and collaborative effect based on holistic attention mechanism. Here is the example of learning the comprehensive preference for local context modeling path, i.e., h^g_{com}, as follows:

$$H_{fea} = concat\left(W^{com}_1 h^g_{short}, W^{com}_2 h_{mul}, W^{com}_3 h^g_{long}\right) \tag{28}$$

$$\xi = softmax\left(H_{fea}\right) \tag{29}$$

$$h^g_{com} = squeeze\_sum\left(\xi H_{fea}\right), \tag{30}$$

where $squeeze\_sum(\cdot)$ denotes a dimension reduction operation based on sum fusion. Unlike traditional fusion methods that use a shallow feed-forward network, the holistic attention mechanism dynamically assigns weights to various preferences, reducing the risk of over-reliance on any single preference. Similarly, the comprehensive preference for local context modeling is denoted by h^{se}_{com}.

4.4. Prediction layer

To predict the next item, the prediction layer provides the recommended score for each candidate item by multiplying the representations of user preference and item as follows:

$$y_i = \frac{exp\left(sim\left(\left(h^g_{com} + h^{se}_{com}\right), e_i\right)/\tau\right)}{\sum_{v_j \in S} exp\left(sim\left(\left(h^g_{com} + h^{se}_{com}\right), e_j\right)/\tau\right)}, \tag{31}$$

where y_i denotes the recommendation score for item v_i, τ denotes the temperature hyper-parameter, $sim(\cdot)$ denotes the similarity between the representation of user preference and the candidate item, e.g., cosine similarity.

Considering both tasks provide significant supervised signals for model training, the total loss is

$$loss = (1-\lambda)\, loss_g + \lambda loss_{se}, \tag{32}$$

where λ is a hyper-parameter to balance the weights of different losses, and the losses of local and global context modeling are constructed as follows:

$$loss_g = -log\left\{\frac{exp\left(sim\left(h^g_{com}, e_+\right)/\tau\right)}{\sum_{v_j\in S} exp\left(sim\left(h^g_{com}, e_j\right)/\tau\right)}\right\}, \tag{33}$$

$$loss_{se} = -log\left\{\frac{exp\left(sim\left(h^{se}_{com}, e_+\right)/\tau\right)}{\sum_{v_j\in S} exp\left(sim\left(h^{se}_{com}, e_j\right)/\tau\right)}\right\}, \tag{34}$$

and e_+ denotes the representation of ground-truth next item v_+ for session S. To mitigate overfitting, we apply dropout to candidate item encoding (Hou et al., 2022). The normalized temperature-scaled cross-entropy loss (Chen et al., 2020) incorporates cosine similarity to enhance feature alignment and uniformity, which helps the model learn more discriminative feature representations. The temperature hyperparameter scales similarity between samples, ensuring consistent importance for comparisons across batches.

4.5. Time complexity analysis

The time complexity of the proposed Coase stems from four key components: local contextual feature extractor, global contextual feature extractor, session-level co-learning module, and prediction layer. For the local contextual feature extractor, the time complexity primarily comes from Bi-gated GSAN. For each layer of Bi-gated GSAN, the time complexity for message passing and node feature fusion involving the neighbor nodes and the central nodes are respectively $O(ud^2+(N+1)|\mathcal{E}|d)$ and $O(2|\mathcal{E}\parallel S|(d^2+d))$, where $|\mathcal{E}|$ denotes the edge number in the session star graph and $|S|$ denotes the number of sessions. Therefore, the overall time complexity of the local contextual feature extractor is $O(L_{gnn}(2(|\mathcal{E}\parallel S|+u)d^2+|\mathcal{E}|(2|S|+N+1)d))$, where L_{gnn} denotes the layer number of Bi-gated GSAN. For the global contextual feature extractor, the time complexity primarily comes from SudokuFormer. For each layer of SudokuFormer, the time complexity for the linear projections of the query, key, and value is $O(7md^2)$, while the time complexity of attention score calculation is $O(10m^2d)$. For the MLP, the time complexity is $O(8md^2)$. Therefore, the overall time complexity of the global contextual feature extractor is $O(L_{trm}(15md^2+10m^2d))$, where L_{gnn} denotes the layer number of SudokuFormer. For the session-level co-learning module, the time complexity of the long-term preference learning for local and global context modeling is $O(2(3md^2+7md))$, while the time complexity of the preference fusion is $O(10d)$. Therefore, the time complexity of the whole session-level co-learning module is $O(6md^2+(14m+10)d)$. Moreover, the time complexity of prediction layer is $O(ud)$. Considering all the above modules, the overall time complexity of Coase is $O((2L_{gnn}(|\mathcal{E}\parallel S|+u)+15L_{trm}m+6m)d^2+(L_{gnn}|\mathcal{E}|(2|S|+N+1)+10L_{trm}m^2+(14m+10)+u)d)$, which remains at an acceptable level.

5. Experiments

In this section, we evaluate the effectiveness of the Coase model by exploring several key research questions: How does Coase compare to 20 baseline models across four real-world datasets (RQ1)? What impact does the collaborative parallel framework have on performance (RQ2)? How do the components of the local contextual feature extractor and SudokuFormer contribute to recommendation performance (RQ3, RQ5)? What effect do temporal elements and hyperparameter settings have on Coase's performance (RQ4, RQ6)? How does Coase perform in sessions of different lengths (RQ7)? How efficient is Coase compared to representative baseline models (RQ8)?

5.1. Datasets

Four publicly available datasets are used in our experiments:

- **Yoochoose** is a dataset from RecSys Challenge 2015.[2] It contains a collection of sessions from an online retailer in Europe during several months in the year of 2014. Each session is encapsulating the click events that the user performed in the session. Due to the large size of Yoochoose, we extracted the most recent 1/64 of the dataset based on the timestamp following Pan et al. (2020), Qiu, Huang, Chen, and Yin (2022), Qiu et al. (2020), and Wu et al. (2019).
- **Diginetica** is a dataset from CIKM Cup 2016.[3] The dataset contains user sessions extracted from an e-commerce search engine log. The transaction data of the dataset is adopted to conduct our experiments.
- **Retailrocket** is a dataset from Kaggle Competition 2016.[4] It was collected from an e-commerce platform within 4.5 months. There are three types of user behaviors in Retailrocket, i.e., view, add to cart, and transaction. The view data of the dataset is adopted to conduct our experiments.
- **Dressipi** is a dataset from RecSys Challenge 2022 focused on fashion recommendation.[5] Dressipi contains training data of 1 million sessions within a month, which is collected from an e-commerce platform.

[2] https://recsys.acm.org/recsys15/challenge
[3] http://cikm2016.cs.iupui.edu/cikm-cup
[4] https://www.kaggle.com/retailrocket/ecommerce-dataset
[5] https://www.recsyschallenge.com/2022

Table 3
Summary of the used datasets.

Dataset	# Sessions	# Items	# Interactions	Avg. session length	Avg. action per item	Sparsity
Yoochoose 1/64	124 724	17 606	528 858	4.24	30.04	99.9759%
Diginetica	204 062	42 172	989 204	4.85	23.46	99.9885%
Retailrocket	328 904	58 000	1 413 976	4.30	24.38	99.9926%
Dressipi	691 383	19 343	4 428 574	6.41	228.96	99.9669%

Following related studies (Chen & Wong, 2020; Hou et al., 2022; Li et al., 2017; Liu et al., 2018; Pan, Cai, Chen, & Chen, 2022; Pan et al., 2020; Qiu, Huang, Chen, & Yin, 2022; Qiu et al., 2020; Wu et al., 2019; Xu et al., 2019), sessions longer than 1 and items appearing more than 4 times are reserved in all the datasets. Table 3 shows the summary statistics for the used four datasets. For fair comparison, the data augment method (Tan et al., 2016) is adopted which generates the sessions and corresponding labels by splitting the input session. For example, for an input session $S = \{v_1, \ldots, v_{|S|}\}$, the generated sessions and the corresponding labels are $(\{v_1\}, v_2), (\{v_1, v_2\}, v_3), \ldots, (\{v_1, \ldots, v_{|S|-1}\}, v_{|S|})$. Moreover, leave-one-out strategy is adopted to split datasets. Specifically, we preserve the last and the second last interactions in each session as the testing and validation data, while the rest is taken as the training data.

5.2. Evaluation metrics

Three metrics are used to evaluate the performance of the model for the top recommended items, with size 5, 10, 15, and 20 in our experiments:

- **Recall@N** assesses the proportion of cases in which the correct items are recommended within the top item list:

$$Recall@N = \frac{1}{M}\sum_{i=1}^{M} hit(i), \tag{35}$$

  where M denotes the number of sessions in test set, $hit(i)$ denotes whether the candidate item i is the correct recommendation so $hit(i) = 1$ if correct otherwise $hit(i) = 0$.
- **MRR@N** measures the average of the reciprocal ranks of the top-ranked relevant item in the recommendations:

$$MRR@N = \frac{1}{M}\sum_{m\in M}\sum_{rank(i)<N}\frac{1}{rank(i)}, \tag{36}$$

  where $rank(i)$ denotes the position of recommended item i in the top item list.
- **NDCG@N** takes into account both the relevance and the position of the recommended items:

$$NDCG@N = \frac{1}{M}\sum_{m\in M}\sum_{i=1}^{N}\frac{2^{rel(i)}-1}{\log_2(1+i)}, \tag{37}$$

  where $rel(i)$ denotes the relevance of the recommended item at position i in the top item list.

5.3. Baseline models and implementation settings

Table 4 summarizes the baseline models, which can be broadly categorized into four groups: traditional methods, local-oriented methods, global-oriented methods, and dual-oriented methods. Our proposed Coase and all the baseline models are implemented based on the popular recommendation framework RecBole (Zhao et al., 2021) and its extension RecBole-GNN (Zhao et al., 2022) for easy development and reproduction. Following Peintner et al. (2023), and Wu et al. (2019), the embedding dimension and batch size is set to 100. The initial learning rate is set to 0.001 and will decay by 10% after each 3 epochs. Following Yang et al. (2023), AdamW optimizer (Loshchilov & Hutter, 2019) is adopted to train the parameters. Following Wang and Liu (2021), the temperature parameter is set to 0.07. To alleviate the overfitting problem, the dropout strategy with 20% ratio has been applied to our model. The max length of session is set to 20. The attention head number of bi-gated graph self-attention network and SudokuFormer is searched among {1, 2, 4, 5}, respectively. The stacking layer number of the bi-gated graph self-attention network and SudokuFormer is searched among {1, 2, 3, 4}, respectively. The loss weight λ is searched among {0, 0.3, 0.5, 0.7, 1}. The number of time intervals is searched among {8, 16, 32, 64}. For all baseline models, we follow the best parameter settings presented in the original papers in most cases. If the parameter configurations are found to be infeasible or unsuitable for our experiment environment, e.g., gradient exploding and out-of-memory errors, we tune the parameters to ensure effective performance. The best results of all baseline models are recorded. For the reader's reference, the implementation of our Coase model is publicly shared at https://anonymous.4open.science/r/Coase-CA97.

Table 4
Summary of the baseline models.

Category	Method	Description
Traditional methods	POP	POP is a non-neural approach recommending the most popular items across the entire training set. Despite its simplicity, POP often serves as a formidable benchmark.
	Item-KNN (Sarwar et al., 2001)	Item-KNN recommends items similar to the last session item based on cosine similarity between binary item vectors.
Local oriented methods	NextItNet (Yuan et al., 2019)	NextItNet is the CNN-based method that adopts dilated convolutions to increase receptive fields instead of suboptimal pooling operation.
	SRGNN (Wu et al., 2019)	SRGNN transforms session sequences into session graphs and applies graph gated neural network to capture pairwise item transition relations.
	GCSAN (Xu et al., 2019)	GCSAN enhances SRGNN by adopting self-attention for capturing long-range item dependencies.
	TAGNN (Yu et al., 2020)	TAGNN strengthens SRGNN with a target-aware attention network to generate user preferences tailored to different candidate items.
	LESSR (Chen & Wong, 2020)	LESSR addresses information loss in GNN-based SBRSs with lossless edge-order preserving aggregation and shortcut graph attention.
Global oriented methods	GRU4Rec (Hidasi et al., 2016)	GRU4Rec stacks multiple gated recurrent unit (GRU) layers to encode the session sequence into a final state. It also applies the ranking loss to train the model.
	NARM (Li et al., 2017)	NARM is the RNN-based method using GRUs for sequential signal capture and attention for emphasizing user intent.
	STAMP (Liu et al., 2018)	STAMP replaces all RNN encoders with attention layers, focusing on the representation of the last item in the session. It does not use any kind of positional encoding.
	STAR (Yeganegi et al., 2024)	STAR improves recommendations by leveraging temporal information in session data, embedding items based on co-occurrence in sub-sessions, and adjusting item weights according to time intervals between events.
	RepeatNet (Ren et al., 2019)	RepeatNet captures the repeat-explore recommendation intent in a session by incorporating a repeat-explore mechanism into RNNs.
	CORE (Hou et al., 2022)	CORE is a simple yet effective framework for session-based recommendation that maintains a consistent representation space throughout encoding and decoding, addressing the issue of inconsistent predictions.
	SASRec (Kang & McAuley, 2018)	SASRec is the self-attention based sequential recommender capturing long-term semantic feature.
	TiSASRec (Li et al., 2020)	TiSASRec improves upon SASRec by incorporating both position information and time interval information.
	AC-TSR (Zhou et al., 2023)	AC-TSR calibrates unreliable attention weights from existing Transformer-based models. Here we adopt the TiSASRec-based version.
	CL4SRec (Xie et al., 2022)	CL4SRec leverages contrastive learning with various data augment methods to capture item content-collaborative signal dependencies.
	DuoRec (Qiu, Huang, Yin, & Wang, 2022)	DuoRec introduces the contrastive regularization to reshape sequence representation distributions.
Dual oriented methods	TCP-SRec (Tian et al., 2022)	TCP-SRec segments interaction sequences into coherent subsequences based on temporal intervals and defines pretraining objectives accordingly.
	COTREC (Xia, Yin, Yu, Shao, et al., 2021)	COTREC is a self-supervised graph co-training framework that iteratively selects evolving pseudo-labels as informative self-supervision examples for enhancing session-based recommendations.
	SGNN-HN (Pan et al., 2020)	SGNN-HN extends SRGNN by introducing session star graph for long-distance information exploration and mitigates over-fitting based on highway networks.
	GCEGNN (Wang et al., 2020)	GCEGNN proposes a unified model that leverages both global and session-level transition patterns between items within global and session graphs.
	CMGNN (Wang, Gao, et al., 2023)	CMGNN is a novel contrastive multi-level graph neural network that captures complex and high-order item transition information.
	FEARec (Du, Yuan, Zhao, Qu, et al., 2023)	FEARec is the contrastive learning based model adopted time domain attention and auto-correlation.
	SLIME4Rec (Du, Yuan, Zhao, Fang, et al., 2023)	SLIME4Rec proposes the dynamic frequency selection and the static frequency split module to capture user dynamic preferences.

5.4. Overall performance

Table 5 presents the average results of our proposed Coase and baseline models in terms of Recall, MRR, and NDCG with various length of top items recommendation list. For the reader's reference, details for each dataset are provided in Appendix. For significance testing, we used a paired t-test with p-value < 0.05. To address RQ1, we have the following observations.

Traditional methods, such as Pop and Item-KNN (Sarwar et al., 2001), disregard contextual information within sessions, leading to suboptimal performance compared to other groups of methods. The naive patterns captured by these traditional approaches are insufficient for representing complex personalized preferences. Furthermore, methods that incorporate temporal information, such as GRU4Rec (Hidasi et al., 2016) and SRGNN (Wu et al., 2019), generally outperform those that neglect it, like STAMP (Liu

Table 5
Summary of the average performance of examined models across four datasets. The best-performing score in each case is highlighted in bold, while the second-best is underlined. A superscript * indicates that Coase significantly outperforms the second-best result, based on a paired t-test with a p-value < 0.05.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Traditional methods	1.83	0.97	1.18	2.95	1.13	1.54	3.78	1.20	1.76	4.73	1.25	1.98
Item-KNN		17.26	10.55	12.08	24.11	11.46	14.30	28.34	11.80	15.42	31.33	11.97	16.13
NextItNet	Local oriented methods	31.31	20.00	22.81	40.36	21.21	25.73	45.64	21.63	27.13	49.31	21.83	28.00
SRGNN		37.72	24.65	27.90	47.21	25.92	30.97	52.51	26.34	32.38	56.16	26.54	33.24
GCSAN		38.72	25.45	28.76	48.14	26.72	31.81	53.40	27.13	33.20	57.04	27.34	34.06
TAGNN		39.32	25.64	29.04	48.85	26.92	32.13	54.15	27.41	33.54	57.76	27.54	34.39
LESSR		38.16	24.95	28.24	47.61	26.22	31.29	52.82	26.63	32.68	56.42	26.83	33.53
GRU4Rec	Global oriented methods	37.33	24.45	27.65	46.84	25.72	30.73	52.19	26.14	32.15	55.87	26.35	33.02
NARM		37.62	24.63	27.86	47.24	25.92	30.97	52.64	26.35	32.41	56.29	26.55	33.27
STAMP		35.08	23.57	26.43	43.84	24.75	29.27	48.91	25.15	30.61	52.48	25.35	31.46
STAR		36.10	24.90	27.88	43.90	26.19	30.40	48.35	26.54	31.58	51.37	26.71	32.29
RepeatNet		37.36	24.80	27.93	45.24	25.86	30.49	49.50	26.19	31.61	52.37	26.36	32.30
CORE		37.62	24.31	27.63	47.09	25.58	30.69	52.45	26.01	32.11	56.04	26.21	32.96
SASRec		37.65	24.99	28.13	47.31	26.28	31.26	52.83	26.72	32.72	56.62	26.93	33.62
TiSASRec		38.87	25.52	28.84	48.46	26.81	31.95	53.90	27.24	33.39	57.62	27.45	34.27
AC-TSR		36.93	24.69	27.74	46.40	25.95	30.80	51.85	26.39	32.24	50.84	26.60	33.14
CL4SRec		35.71	23.85	26.80	44.76	25.06	29.73	50.11	25.49	31.14	53.81	25.69	32.02
DuoRec		36.93	24.71	27.74	46.18	25.94	30.74	51.61	26.37	32.17	55.37	26.58	33.06
TCP-SRec	Dual oriented methods	32.65	21.40	24.19	41.51	22.59	27.06	46.69	23.00	28.44	50.30	23.20	29.29
COTREC		31.06	24.67	26.27	34.71	25.14	27.42	36.84	25.30	27.96	38.43	25.39	28.32
SGNN-HN		38.81	25.09	28.50	48.46	26.39	31.63	53.83	26.81	33.06	57.51	27.02	33.93
GCEGNN		39.66	25.80	29.25	49.19	27.08	32.34	54.52	27.50	33.75	58.14	27.71	34.61
CMGNN		39.06	25.37	28.77	48.71	26.66	31.90	54.01	27.08	33.30	57.68	27.29	34.17
FEARec		37.08	24.76	27.83	46.27	25.99	30.80	51.61	26.41	32.21	55.30	26.62	33.09
SLIME4Rec		39.03	25.73	29.04	48.73	27.03	32.18	54.21	27.46	33.63	57.97	27.67	34.52
Coase (Ours)		**40.35**	**26.53***	**29.96***	**50.09***	**27.83***	**33.12***	**55.49***	**28.25***	**34.55***	**59.13***	**28.46***	**35.41***
Improvement		1.73%	2.80%	2.43%	1.83%	2.77%	2.40%	1.79%	2.73%	2.36%	1.71%	2.73%	2.32%

et al., 2018) and NextItNet (Yuan et al., 2019). This underscores the importance of temporal information in understanding behavior patterns and effectively modeling user preferences within a session.

It has been observed that SRGNN (Wu et al., 2019) outperforms NextItNet (Yuan et al., 2019), indicating that the session graph contains more local contextual information than the raw session sequence. Specifically, the session graph provides a unique perspective to represent both direct and indirect transition relationships between items, while the session sequence retains only simplistic sequential transition relationships. Additionally, most transformer-based methods (Kang & McAuley, 2018; Li et al., 2020) consistently outperform RNN-based methods (Hidasi et al., 2016; Li et al., 2017), highlighting the superiority of transformers in capturing item dependencies within sequential data. Furthermore, TiSASRec (Li et al., 2020) consistently outperforms SASRec (Kang & McAuley, 2018), demonstrating the benefits of incorporating various temporal information. In other words, users' choices regarding items are influenced not only by their long-term and short-term preferences but also by time-sensitive contexts (Wang, Cao, et al., 2022). However, the performance of AC-TSR (Zhou et al., 2023), which builds upon TiSASRec (Li et al., 2020), falls short compared to its base model. This discrepancy underscores the importance of effectively managing temporal information in session-based recommendation tasks.

Most dual-oriented methods outperform the other three groups, demonstrating the effectiveness of integrating both local and global context modeling. In particular, our Coase approach exhibits superior performance compared to many baseline methods across various metrics. The improvements can be categorized into three key aspects. Firstly, Coase utilizes the session star graph to illustrate both direct and indirect transitions among items, effectively addressing the challenge of long-range information propagation. By employing a bi-gated graph self-attention network on this graph, Coase enhances its ability to capture local contextual features. Secondly, Coase incorporates both positional and time interval information through explicit encoding methods, enriching the characterization of user behaviors. Position encoding helps to understand the order of interactions, while time interval encoding reveals the level of user interest in items. Additionally, the SudokuFormer architecture accommodates data heterogeneity, providing the flexibility and expressive power needed to adaptively learn item representations. It computes disentangled attention weights and employs a stable, non-invasive fusion method to capture global contextual information. Lastly, Coase introduces a triple attention mechanism to learn diverse session-level representations. This mechanism considers the collaborative effects between local and global context modeling, enabling the model to effectively capture comprehensive behavioral patterns.

5.5. Ablation studies

Four groups of ablation studies are conducted to assess the effectiveness of key components in Coase by comparing it with its variants. These studies evaluate the unified collaborative parallel framework, the local context modeling components, the impact of various encodings for global context modeling, and each component of the proposed SudokuFormer.

Table 6
Effects of unified collaborative parallel framework. For each dataset, the **bold**-faced number is the best score.

Models	Graph learning	Sequential learning	Collaborative effect	@5			@10			@15			@20		
				Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64															
Coase	✔	✔	✔	**49.05**	**30.45**	**35.07**	**61.25**	**32.09**	**39.03**	**67.12**	**32.55**	**40.59**	**70.50**	**32.75**	**41.39**
w/o local context modeling	✖	✔	✔	48.91	30.33	34.95	60.85	31.94	38.82	66.60	32.39	40.35	70.00	32.58	41.15
w/o global context modeling	✔	✖	✔	48.50	30.13	34.70	60.65	31.77	38.64	66.32	32.22	40.15	69.84	32.42	40.98
w/o collaborative effect	✔	✔	✖	**49.05**	30.41	35.05	61.21	32.05	38.99	66.87	32.50	40.50	70.28	32.69	41.30
Degenerate learning and fusion	✔	✔	✖	48.76	30.39	34.96	60.73	32.00	38.84	66.37	32.45	40.34	69.77	32.64	41.14
Diginetica															
Coase	✔	✔	✔	**31.79**	**18.75**	**21.98**	**43.33**	**20.29**	**25.71**	**50.46**	**20.85**	**27.60**	**55.53**	**21.14**	**28.80**
w/o local context modeling	✖	✔	✔	31.58	18.68	21.87	42.94	20.18	25.54	50.14	20.75	27.45	55.18	21.04	28.64
w/o global context modeling	✔	✖	✔	31.06	18.08	21.30	42.50	19.60	24.99	49.57	20.16	26.86	54.67	20.44	28.07
w/o collaborative effect	✔	✔	✖	31.58	18.54	21.77	42.93	20.05	25.44	50.08	20.62	27.33	55.17	20.90	28.53
Degenerate learning and fusion	✔	✔	✖	29.81	16.66	19.91	41.67	18.24	23.74	49.05	18.82	25.69	54.30	19.11	26.93
Retailrocket															
Coase	✔	✔	✔	**56.90**	**42.91**	**46.42**	**63.73**	**43.83**	**48.63**	**67.16**	**44.10**	**49.54**	**69.47**	**44.23**	**50.09**
w/o local context modeling	✖	✔	✔	54.57	40.72	44.19	61.70	41.68	46.50	65.49	41.98	47.50	67.95	42.12	48.09
w/o global context modeling	✔	✖	✔	54.89	40.07	43.78	62.49	41.10	46.25	66.51	41.41	47.32	69.15	41.56	47.94
w/o collaborative effect	✔	✔	✖	55.06	40.80	44.37	62.53	41.81	46.80	66.34	42.11	47.80	68.96	42.25	48.42
Degenerate learning and fusion	✔	✔	✖	55.05	40.72	44.31	62.51	41.73	46.73	66.35	42.04	47.75	68.89	42.18	48.35
Dressipi															
Coase	✔	✔	✔	**23.65**	**13.99**	**16.38**	**32.03**	**15.11**	**19.09**	**37.23**	**15.51**	**20.46**	**41.03**	**15.73**	**21.36**
w/o local context modeling	✖	✔	✔	22.21	13.13	15.38	30.31	14.21	18.00	35.38	14.61	19.34	39.10	14.82	20.22
w/o global context modeling	✔	✖	✔	21.32	12.39	14.60	29.38	13.46	17.21	34.48	13.87	18.55	38.23	14.08	19.44
w/o collaborative effect	✔	✔	✖	20.89	12.21	14.36	28.68	13.25	16.87	33.61	13.63	18.18	37.33	13.84	19.06
Degenerate learning and fusion	✔	✔	✖	20.25	12.06	14.09	27.56	13.03	16.45	32.14	13.39	17.66	35.60	13.59	18.48

5.5.1. Impact of unified collaborative parallel framework

The first group of ablation studies addresses RQ2 and demonstrates the effectiveness of the unified collaborative parallel framework in Coase. Specifically, we generate four variants for comparison: (i) w/o local context modeling, which removes the session star graph and bi-gated graph self-attention network for local context modeling; (ii) w/o global context modeling, which excludes the dual global position encoding and SudokuFormer used for global context modeling; (iii) w/o collaborative effect, which learns long-term and comprehensive preferences without considering the collaborative effect; and (iv) Degenerate learning and fusion, which learns short-term and long-term preferences similar to SRGNN (Wu et al., 2019), then fuse both preferences by naive concat operation. The results, shown in Table 6, consistently indicate a performance drop when any part of the collaborative parallel framework is removed, highlighting the importance of local context modeling, global context modeling, and the collaborative effect in capturing comprehensive preferences. In particular, the w/o global context modeling variant underperforms Coase, underscoring the critical role of SudokuFormer in learning user preferences from a global context. Similarly, the w/o local context modeling variant also underperforms, suggesting that non-sequential transitions between items are crucial for capturing user preferences. Notably, the performance drop in w/o global context modeling is more pronounced than in w/o local context modeling, demonstrating that global contextual features are more significant in session-based recommendation tasks. This finding reveals that user preferences exhibit discernible continuity and coherence within sessions, implying that user behaviors in both datasets follow predictable temporal patterns. As a result, understanding user preferences through global context modeling proves more advantageous.

Comparing to the w/o collaborative effect variant and the degenerate learning and fusion variant, our Coase achieves better performance in the most cases, demonstrating the effectiveness of the preference learning methods and the preference fusion method in the session-level co-learning module. We also observe that the w/o collaborative effect variant performs worse than Coase in larger datasets (Retailrocket and Dressipi), indicating that the collaborative effect refines user preference representation. However, in the two smaller datasets (Yoochoose 1/64 and Diginetica), the coase model only achieved a slight advantage over the w/o collaborative effect variant. To demonstrate the effectiveness of our model design, we conduct statistical significance tests by comparing our model against the w/o collaborative effect variant in the two smaller datasets. Similar to the settings of the baseline experiments, we conducted paired t-tests using five different random seeds to evaluate the statistical significance of our improvements. The results of these additional experiments show that Coase significantly outperforms the w/o collaborative effect variant with a p-value < 0.05. It indicates that while the performance advantage of our proposed method is subtle on smaller datasets, it is statistically significant. One reason is that the local contextual feature extractor and the global contextual feature extractor tend to learn similar patterns due to the limited diversity in user interactions within smaller datasets. Therefore, the simple preference learning method can achieve acceptable performance. However, as the dataset size increases, user behavior patterns become more diverse. In these cases, if the local and global contextual feature extractors operate independently, they may capture significantly different user behavioral patterns. When applied to larger datasets, the simple method leads to feature conflicts, as the differences between locally and globally learned features become more pronounced, resulting in a substantial drop in performance.

5.5.2. Impact of local context modeling components

The second group of ablation studies addresses RQ3, demonstrating the effectiveness of the session star graph and the Bi-Gated GSAN in local context modeling. We compare four variants: (i) GAT with SG on SSG, which removes the gating network and updates

Table 7
Effects of components in local context modeling. For each dataset, the **bold**-faced number is the best score.

Models	Dual gate	Message passing	Star session graph	@5			@10			@15			@20		
				Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64															
Coase	✔	Attention	✔	**49.05**	**30.45**	**35.07**	**61.25**	**32.09**	**39.03**	**67.12**	**32.55**	**40.59**	**70.50**	**32.75**	**41.39**
GAT with SG on SSG	✖	Attention	✔	48.29	29.82	34.42	60.25	31.43	38.29	66.15	31.90	39.86	69.56	32.09	40.67
GGNN with SG on SSG	✖	GRU	✔	48.67	30.16	34.77	60.82	31.79	38.71	66.50	32.24	40.21	69.98	32.44	41.03
GAT with SG on VSG	✖	Attention	✖	48.38	29.82	34.43	60.67	31.47	38.42	66.56	31.94	39.98	70.03	32.14	40.80
GGNN with SG on VSG	✖	GRU	✖	48.99	30.37	35.00	61.19	32.01	38.96	66.87	32.46	40.47	70.34	32.66	41.29
Diginetica															
Coase	✔	Attention	✔	**31.79**	**18.75**	**21.98**	**43.33**	**20.29**	**25.71**	**50.46**	**20.85**	**27.60**	**55.53**	**21.14**	**28.80**
GAT with SG on SSG	✖	Attention	✔	30.87	18.23	21.36	42.04	19.72	24.97	48.94	20.26	26.80	53.90	20.54	27.97
GGNN with SG on SSG	✖	GRU	✔	**31.79**	18.62	21.89	43.25	20.15	25.59	50.34	20.71	27.46	55.44	20.99	28.67
GAT with SG on VSG	✖	Attention	✖	31.22	18.47	21.63	42.56	19.98	25.29	49.62	20.54	27.16	54.70	20.82	28.36
GGNN with SG on VSG	✖	GRU	✖	31.71	18.23	21.36	43.14	19.72	24.97	50.23	20.26	26.80	55.30	20.54	27.97
Retailrocket															
Coase	✔	Attention	✔	**56.90**	**42.91**	**46.42**	**63.73**	**43.83**	**48.63**	**67.16**	**44.10**	**49.54**	**69.47**	**44.23**	**50.09**
GAT with SG on SSG	✖	Attention	✔	54.38	40.14	43.71	61.72	41.13	46.09	65.69	41.45	47.14	68.27	41.59	47.75
GGNN with SG on SSG	✖	GRU	✔	55.10	40.73	44.32	62.56	41.73	46.75	66.27	42.03	47.73	68.79	42.17	48.33
GAT with SG on VSG	✖	Attention	✖	54.85	40.53	44.12	62.10	41.51	46.47	66.06	41.82	47.52	68.64	41.97	48.13
GGNN with SG on VSG	✖	GRU	✖	55.08	40.66	44.27	62.52	41.66	46.69	66.46	41.97	47.73	68.95	42.11	48.32
Dressipi															
Coase	✔	Attention	✔	**23.65**	**13.99**	**16.38**	**32.03**	**15.11**	**19.09**	**37.23**	**15.51**	**20.46**	**41.03**	**15.73**	**21.36**
GAT with SG on SSG	✖	Attention	✔	21.67	12.59	14.84	29.77	13.67	17.45	34.92	14.07	18.81	38.67	14.28	19.70
GGNN with SG on SSG	✖	GRU	✔	21.92	12.83	15.08	29.98	13.90	17.68	35.05	14.30	19.03	38.82	14.52	19.92
GAT with SG on VSG	✖	Attention	✖	21.65	12.65	14.88	29.84	13.74	17.52	34.97	14.15	18.88	38.74	14.36	19.77
GGNN with SG on VSG	✖	GRU	✖	22.34	13.14	15.42	30.54	14.23	18.07	35.64	14.63	19.42	39.39	14.84	20.30

the item features based solely on neighboring items; (ii) GGNN with SG on SSG, which replaces the graph self-attention network (GAT) with a gated graph neural network (GGNN); (iii) GAT with SG on VSG, which replaces the session star graphs with vanilla session graphs compared to (i); and (iv) GGNN with SG on VSG, which replaces the GAT with GGNN compared to (iii). Table 7 presents the results, leading to several key observations. In most cases, GGNN-based methods outperform GAT-based ones, suggesting that gating mechanisms are effective in filtering out noisy features from neighboring nodes. Additionally, the informative session star graph does not consistently enhance performance, and GGNN appears to be more compatible with this structure compared to vanilla GAT. The virtual central node establishes bidirectional connections with each item node, thereby improving the connectivity of the session graph and facilitating message passing among item nodes. However, treating the central node as an ordinary item node can introduce excessive contextual information, potentially overwhelming the target item node—particularly in GAT-based models. This highlights the need for a tailored gating mechanism to selectively absorb information from the central node. Such an approach mitigates the influence of noisy features originating from distant item nodes, preserving the integrity of the target item's representation. Finally, Coase consistently outperforms all four variants. This can be attributed to its bi-gated graph self-attention network, which enhances vanilla GAT by integrating two gating mechanisms that effectively filter and select relevant features from different nodes.

5.5.3. Impact of encodings for global context modeling

The third group of ablation studies addresses RQ4, aiming to demonstrate the effectiveness of different encodings in global context modeling. Specifically, we produce two variants for comparison: (i) w/o time interval encoding, which removes the time interval encoding as the input of our SudokuFormer; and (ii) w/o position encoding, which removes the position encoding as the input of our SudokuFormer. The results, shown in Table 8, indicate that both variants perform worse than Coase, underscoring the effectiveness of both position and time interval encodings. The core of SudokuFormer is a multi-head disentangled self-attention mechanism, a variant of the vanilla self-attention network. While powerful in capturing dependencies between elements in a sequence, the vanilla self-attention network lacks positional awareness due to permutation equivalence (Liu et al., 2020; Raffel et al., 2020). This means it treats all elements equally, regardless of their position, which can result in sub-optimal performance. In session-based recommendation tasks, permutation equivalence implies that the order of items could be randomized without affecting the learned features, which contradicts the assumption that the order of items is crucial for capturing user preferences. While position encoding is useful for understanding the order of interactions, it does not reveal the level of interest in each interaction. Time interval encoding addresses this by parameterizing time cues, providing fine-grained insights into how user interest evolves over time. As a result, both position and time interval encodings are essential for learning comprehensive dynamic preferences. Furthermore, the results show that w/o time interval encoding performs worse than w/o position encoding in most cases, indicating that the degree of user interest in items (captured through time interval encoding) plays a more critical role than the sequential order of items in understanding behavioral patterns.

Table 8
Effects of encodings in global context modeling. For each dataset, the **bold**-faced number is the best score.

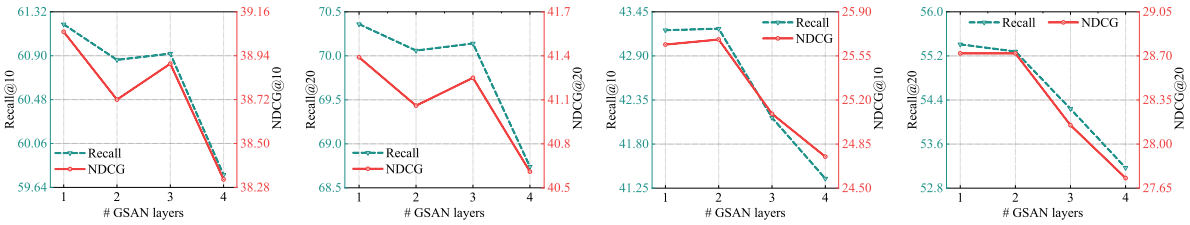
Models	Time interval information	Position information	@5			@10			@15			@20		
			Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64														
Coase	✔	✔	**49.05**	**30.45**	**35.07**	**61.25**	**32.09**	**39.03**	**67.12**	**32.55**	**40.59**	**70.50**	**32.75**	**41.39**
w/o time interval encoding	✖	✔	48.36	30.12	34.66	60.39	31.74	38.57	65.93	32.18	40.04	69.51	32.39	40.88
w/o position encoding	✔	✖	48.84	30.24	34.87	61.06	31.89	38.84	66.91	32.36	40.39	70.20	32.54	41.17
Diginetica														
Coase	✔	✔	**31.79**	**18.75**	**21.98**	**43.33**	**20.29**	**25.71**	**50.46**	**20.85**	**27.60**	**55.53**	**21.14**	**28.80**
w/o time interval encoding	✖	✔	31.39	18.49	21.69	42.92	20.03	25.41	50.05	20.59	27.30	55.20	20.88	28.52
w/o position encoding	✔	✖	**31.79**	18.69	21.93	43.14	20.20	25.60	50.31	20.76	27.50	55.41	21.05	28.70
Retailrocket														
Coase	✔	✔	**56.90**	**42.91**	**46.42**	**63.73**	**43.83**	**48.63**	**67.16**	**44.10**	**49.54**	**69.47**	**44.23**	**50.09**
w/o time interval encoding	✖	✔	54.78	40.57	44.13	62.23	41.58	46.55	66.13	41.88	47.58	68.61	42.02	48.17
w/o position encoding	✔	✖	54.74	40.51	44.07	62.09	41.50	46.46	65.88	41.80	47.47	68.32	41.94	48.04
Dressipi														
Coase	✔	✔	**23.65**	**13.99**	**16.38**	**32.03**	**15.11**	**19.09**	**37.23**	**15.51**	**20.46**	**41.03**	**15.73**	**21.36**
w/o time interval encoding	✖	✔	21.62	12.65	14.87	29.69	13.73	17.48	34.77	14.13	18.82	38.52	14.34	19.71
w/o position encoding	✔	✖	21.90	12.83	15.08	30.04	13.92	17.71	35.12	14.32	19.05	38.86	14.53	19.93

Table 9
Effects of SudokuFormer. For each dataset, the **bold**-faced number is the best score.

Models	Disentangled representation	Non-invasive method	Dual layer norm	@5			@10			@15			@20		
				Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64															
Coase	✔	✔	✔	**49.05**	**30.45**	**35.07**	**61.25**	**32.09**	**39.03**	**67.12**	**32.55**	**40.59**	**70.50**	**32.75**	**41.39**
w/o disentangled representation	✖	✔	✔	48.88	30.33	34.94	60.98	31.96	38.87	66.72	32.41	40.39	70.19	32.61	41.21
w/o non-invasive method	✔	✖	✔	48.84	30.32	34.93	61.08	31.97	38.90	66.76	32.42	40.41	70.24	32.62	41.23
w/o dual layer norm	✔	✔	✖	48.95	30.39	35.01	61.24	32.05	39.00	66.96	32.50	40.52	70.34	32.69	41.32
Diginetica															
Coase	✔	✔	✔	**31.79**	**18.75**	**21.98**	**43.33**	**20.29**	**25.71**	**50.46**	**20.85**	**27.60**	**55.53**	**21.14**	**28.80**
w/o disentangled representation	✖	✔	✔	31.67	18.63	21.86	43.00	20.14	25.52	50.18	20.71	27.42	55.27	20.99	28.62
w/o non-invasive method	✔	✖	✔	31.72	18.72	21.94	43.23	20.25	25.66	50.31	20.81	27.53	55.49	21.10	28.76
w/o dual layer norm	✔	✔	✖	31.41	18.53	21.72	42.89	20.06	25.44	49.91	20.62	27.29	54.97	20.90	28.49
Retailrocket															
Coase	✔	✔	✔	**56.90**	**42.91**	**46.42**	**63.73**	**43.83**	**48.63**	**67.16**	**44.10**	**49.54**	**69.47**	**44.23**	**50.09**
w/o disentangled representation	✖	✔	✔	54.93	40.50	44.11	62.23	41.49	46.48	66.20	41.80	47.54	68.65	41.94	48.12
w/o non-invasive method	✔	✖	✔	54.74	40.44	44.02	62.27	41.46	44.02	66.10	41.76	47.48	68.58	41.76	48.07
w/o dual layer norm	✔	✔	✖	54.93	40.51	44.12	62.41	41.52	46.55	66.38	41.83	47.60	68.95	41.98	48.21
Dressipi															
Coase	✔	✔	✔	**23.65**	**13.99**	**16.38**	**32.03**	**15.11**	**19.09**	**37.23**	**15.51**	**20.46**	**41.03**	**15.73**	**21.36**
w/o disentangled representation	✖	✔	✔	22.01	12.86	15.13	30.16	13.95	17.76	35.32	14.35	19.13	39.11	14.57	20.02
w/o non-invasive method	✔	✖	✔	21.72	12.67	14.91	29.81	13.74	17.52	34.94	14.15	18.87	38.70	14.36	19.76
w/o dual layer norm	✔	✔	✖	21.93	12.81	15.07	30.05	13.89	17.69	35.20	14.30	19.05	38.99	14.51	19.95

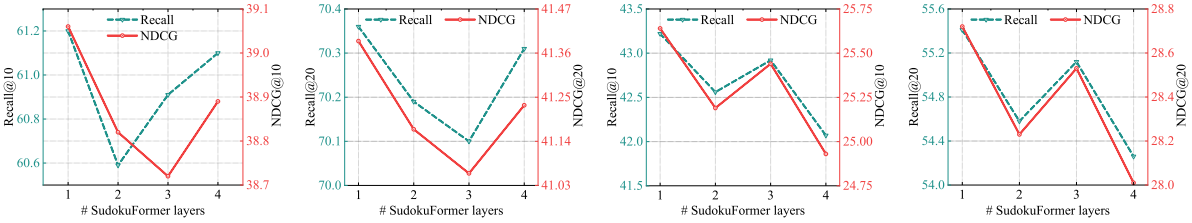
5.5.4. Impact of each component in SudokuFormer

The last group of ablation studies addresses RQ5, aiming to demonstrate the effectiveness of each component in the proposed SudokuFormer. Specifically, we compare three variants: (i) w/o disentangled representation, which combines position and time interval encodings as a single input to SudokuFormer; (ii) w/o non-invasive method, which fuses position and time interval encodings into the Values using a mean pooling operation; and (iii) w/o dual layer norm, which removes the sub-layer norm operations and retains only the original post-layer norm. The results, shown in Table 9, reveal that removing any of these components results in a performance drop, with the full Coase model achieving the best performance, highlighting the importance of the disentangled attention mechanism and the stable non-invasive fusion method. SudokuFormer represents each item using three distinct vectors that encode its content, position, and time cues, which are used to compute disentangled attention weights between items. In the w/o disentangled representation variant, the position and time interval encodings are combined, leading to biased attention matrices that fail to capture their distinct roles in modeling temporal information. Additionally, the non-invasive method treats positional and time interval information as auxiliary features to improve the attention mechanism's distribution, effectively reducing information overload (Liu et al., 2021). Finally, similar to its successful application in natural language processing tasks (Wang, Ma, et al., 2024, 2023), the dual layer norm method enhances the stability of learned attention weights by incorporating additional normalization, which contributes to more robust and stable learning.



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 4. Coase performance with varying Bi-Gated GSAN stacking layers on Yoochoose 1/64 (Yoo) and Diginetica (Digi).



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 5. Coase performance with varying SudokuFormer stacking layers on Yoochoose 1/64 (Yoo) and Diginetica (Digi).

5.6. Parameter sensitivity analysis

To address RQ6, we investigate the impact of key hyperparameters on the performance of Coase, such as the number of stacking layers, attention heads, and the time interval. Additionally, we assess the influence of the loss weight, which balances the contributions of both tasks in the model. Furthermore, we examine the influence of the temperature parameter in the loss function, which controls the strength of penalties on hard negative samples. Finally, we investigate the scalability of Coase by tuning the embedding dimension. For the sake of efficiency, we report results from the Yoochoose 1/64 and Diginetica datasets, while the similar trends are observed in the other two datasets.

5.6.1. Impact of different number of stacking layers

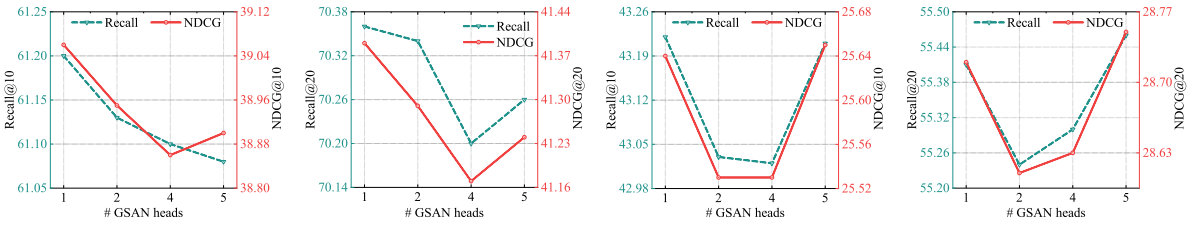
Figs. 4–5 illustrate the impact of the number of stacking layers on model performance, tuned in $\{1, 2, 3, 4\}$ for the bi-gated graph self-attention network and SudokuFormer, respectively. We observe that Coase achieves the best performance with 1 layer, and performance steadily decreases as the number of layers increases. This demonstrates that Coase does not benefit from deeper networks, likely due to the over-smoothing problem, where the representations of different items become too similar as the layers increase, leading to homogenization of information and reduced distinction between item representations.

5.6.2. Impact of different number of attention heads

As shown in Figs. 6–7, we increase the number of attention heads in both the bi-gated graph self-attention network and SudokuFormer, tuning values in the set $\{1, 2, 4, 5\}$, to improve the model's stability and effectiveness. However, this approach proves time-consuming and does not enhance performance—rather, it results in a decline. We attribute this diminished performance to the limited amount of information in the datasets, which may not fully benefit from the multi-head mechanism. While adding more attention heads can theoretically increase model stability and enable learning from diverse sub-spaces, it also leads to inefficiencies due to redundant information, making the mechanism cumbersome and less effective (Yin et al., 2023).

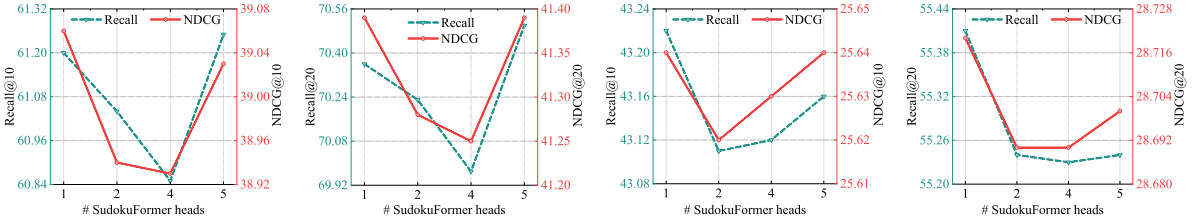
5.6.3. Impact of different number of time intervals

Fig. 8 evaluates the impact of the time interval numbers in $\{8, 16, 32, 64\}$. It is observed that Coase benefits from a larger maximum interval on the Yoochoose 1/64 dataset, while it achieves higher performance with a smaller maximum interval on the Diginetica dataset. Specifically, Coase performs best with 16 or 32 time intervals on Yoochoose 1/64, whereas the most suitable value is 8 on Diginetica. This variation can be attributed to differences in the distribution of time intervals across datasets. Generally, users in the Yoochoose 1/64 dataset tend to make decisions quickly, with rapidly shifting preferences. In contrast, Coase encodes fewer time intervals for Diginetica, where users typically take longer to make decisions.



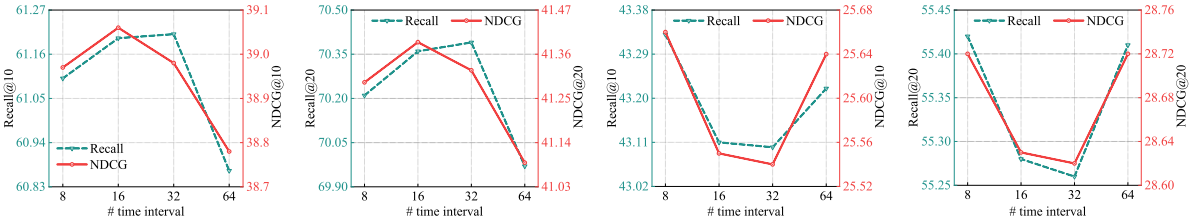
(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 6. Coase performance with varying Bi-Gated GSAN attention heads on Yoochoose 1/64 (Yoo) and Diginetica (Digi).



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 7. Coase performance with varying SudokuFormer attention heads on Yoochoose 1/64 (Yoo) and Diginetica (Digi).



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

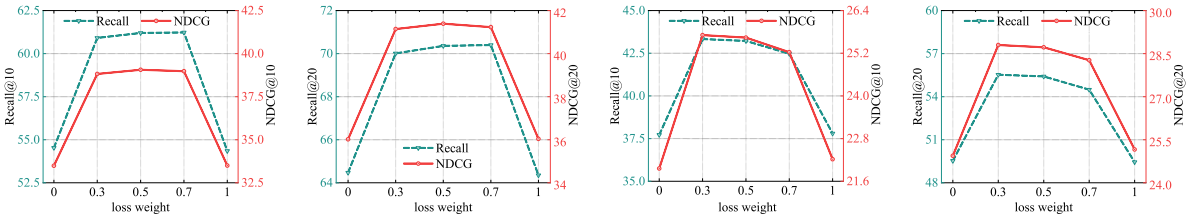
Fig. 8. Coase performance with varying time intervals on Yoochoose 1/64 (Yoo) and Diginetica (Digi).

5.6.4. Impact of different loss weight

In Coase, a hyperparameter is used to balance the two-part cross-entropy loss from local and global context modeling, respectively. As shown in Fig. 9, to demonstrate the contribution of each task, we compare the experimental results by tuning the values from $\{0, 0.3, 0.5, 0.7, 1\}$. We observe that Coase shows worse performance when loss weight is 0 and 1, indicating that both preferences are critical for model training. Furthermore, it is found that Coase achieves its best performance on the Yoochoose 1/64 dataset when the loss weight is 0.5, while the best loss weight is 0.3 on the Diginetica dataset. We attribute the reason that the Diginetica dataset contains rich local contextual information, which enables the local context modeling to offer more supervision signals for training. By contrast, local and global context modeling tend to be equally crucial in learning the user behavioral patterns on the Yoochoose 1/64 dataset.

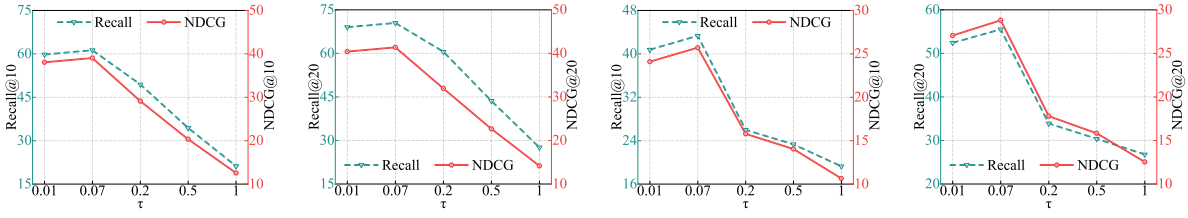
5.6.5. Impact of different temperature parameter

As shown in Fig. 10, we tune τ from $\{0.01, 0.07, 0.2, 0.5, 1\}$ to evaluate its impact. It is observed that the performance improves with an increase in τ across both datasets, reaching its peak at $\tau = 0.07$. However, further increasing τ significantly degrades the performance. It is attributed that the temperature parameter plays a crucial role in regulating the strength of penalties on hard negative samples (Wang & Liu, 2021). Specifically, a small τ tends to impose greater penalties on the hardest negative samples, resulting in a more distinct separation of the local structure around each sample and a more uniform embedding distribution.



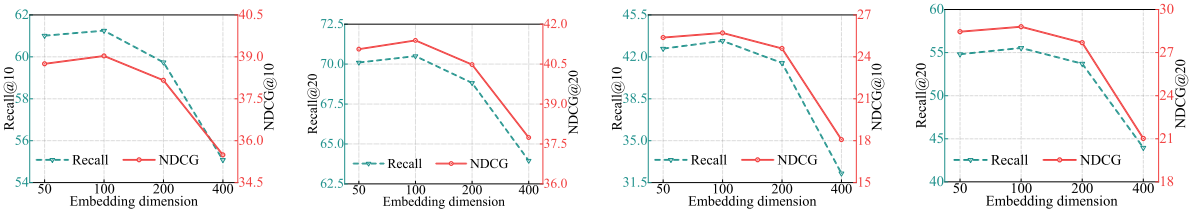
(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 9. Coase performance with varying loss weights on Yoochoose 1/64 (Yoo) and Diginetica (Digi).



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 10. Coase performance with varying temperature parameters on Yoochoose 1/64 (Yoo) and Diginetica (Digi).



(a) Metrics@10 on Yoo. (b) Metrics@20 on Yoo. (c) Metrics@10 on Digi. (d) Metrics@20 on Digi.

Fig. 11. Coase performance with varying embedding dimensions on Yoochoose 1/64 (Yoo) and Diginetica (Digi).

Moreover, a large τ tends is less sensitive to hard negative samples, improving the tolerance to the semantically similar samples. Therefore, fine-tuning the temperature parameter is essential for achieving an optimal balance between uniformity and tolerance.

5.6.6. *Impact of different embedding dimensions*

As shown in Fig. 11, we vary the embedding dimension across $50, 100, 200, 400$ to evaluate the scalability of Coase. The results show that Coase's performance – measured by Recall@10, Recall@20, NDCG@10, and NDCG@20 – improves as the embedding dimension increases from 50 to 100. However, further increasing the embedding size leads to a decline in performance. This suggests that while a smaller embedding dimension may constrain the model's representational capacity, excessively large embeddings may introduce overfitting. These findings demonstrate Coase's scalability and the importance of selecting an appropriate embedding size.

5.7. *Impact of the session length*

The session length is a key factor that influences model performance of a SBRS, since that it signifies how much information the model can rely on to capture user preference. To address RQ7, we explore the performance of our Coase, the two variant models of Coase (w/o local context modeling and w/o global context modeling), and seven representative baseline models, i.e., TAGNN (Yu et al., 2020), SASRec (Kang & McAuley, 2018), SGNN-HN (Pan et al., 2020), NARM (Li et al., 2017), CORE (Hou et al., 2022), GCEGNN (Wang et al., 2020), and CMGNN (Wang, Gao, et al., 2023), under different session lengths in the Diginetica dataset. We categorize sessions into short, middle, and long groups with thresholds of 5 and 15 items (short: 5 or fewer, middle: 5 to 15, long: more than 15). The dataset statistics are presented in Table 10, showing that 71.03% of the sessions are short, 26.94% are middle-length, and 2.03% are long. This distribution reveals that most real-world sessions are short, making it challenging to capture user behavioral patterns due to limited contextual information in short and middle sessions. Despite these challenges, Table 11 shows that Coase outperforms the five baseline models on both the Diginetica-short and Diginetica-middle datasets, while achieving comparable performance on the Diginetica-long dataset. These results highlight the effectiveness of Coase in session-based recommendation tasks, even with limited session lengths.

Table 10
Statistics of Diginetica-short, Diginetica-middle, and Diginetica-long.

Dataset	# Sessions	# Items	# Interactions	Avg. session length	Avg. action per item
Diginetica-short	144 939	40 782	444 678	3.07	10.90
Diginetica-middle	54 984	41 011	462 360	8.41	11.27
Diginetica-long	4141	23 071	82 166	19.85	3.56

Table 11
Model performance on short, middle, and long sessions. For each dataset, the **bold**-faced number is the best score and the second performer is underlined.

Model	Diginetica-short						Diginetica- middle						Diginetica- long					
	@10			@20			@10			@20			@10			@20		
	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Coase	**40.47**	**20.43**	**25.18**	**49.86**	**21.08**	**27.55**	**37.94**	**17.03**	**21.94**	**50.16**	**17.88**	**25.03**	**30.51**	13.94	17.82	**39.25**	14.54	**20.02**
w/o graph learning	40.25	20.15	24.91	49.68	20.80	27.29	37.28	16.83	21.63	49.54	17.69	24.73	29.78	13.97	17.67	37.83	14.52	19.70
w/o sequence learning	40.19	20.25	24.98	49.59	20.91	27.35	37.44	16.52	21.43	49.34	17.35	24.44	30.41	**14.00**	**17.85**	38.70	**14.57**	19.94
TAGNN	36.14	18.11	22.38	44.58	18.70	24.51	34.21	15.23	19.67	45.40	16.00	22.50	24.35	11.52	14.53	32.00	12.06	16.47
SASRec	36.51	18.30	22.58	46.14	18.96	25.01	33.60	14.98	19.33	45.70	15.82	22.39	27.13	12.63	16.03	36.14	13.24	18.30
SGNN-HN	38.80	19.14	23.78	48.24	19.79	26.17	35.48	15.74	20.36	47.14	16.54	23.31	29.11	13.33	17.03	36.79	13.86	18.97
NARM	33.40	15.91	20.03	42.32	16.53	21.41	33.07	14.42	18.78	44.50	15.22	21.67	25.07	11.15	14.42	32.42	11.65	16.27
CORE	39.16	18.85	23.65	49.48	19.57	26.26	35.88	16.47	21.02	47.04	17.24	23.85	29.23	12.93	16.73	38.09	13.54	18.98
GCEGNN	38.86	19.36	23.97	48.51	20.03	26.41	37.04	16.26	21.14	48.94	17.09	24.14	29.03	13.73	17.33	35.92	14.21	19.08
CMGNN	38.10	18.93	23.47	47.17	19.56	25.76	35.62	15.61	20.30	47.31	16.41	23.25	28.55	13.12	16.76	36.50	13.67	18.77

Moreover, compared w/o local context modeling to NARM (Li et al., 2017), SASRec (Kang & McAuley, 2018), and CORE (Hou et al., 2022), we argue that our SudokuFormer benefits from the disentangled attention mechanism and the stable non-invasive fusion method to learn more advanced temporal item transition patterns. Similarly, w/o global context modeling outperforms TAGNN (Yu et al., 2020) demonstrates that our Bi-Gated GSAN captures representative non-sequential item dependencies by adopting bi-gated graph self-attention network on session star graphs. Furthermore, it is found that the local–global context modeling-based methods achieves better performance than the single learning-based methods in most cases. Specifically, SGNN-HN (Pan et al., 2020), GCEGNN (Wang et al., 2020), and CMGNN (Wang, Gao, et al., 2023) consistently outperforms TAGNN (Yu et al., 2020), NARM (Li et al., 2017), and SASRec (Kang & McAuley, 2018). Meanwhile, our Coase outperforms its two variant models in most cases. It can be attributed that the features derived from local and global context modeling are complementary, thus the learning process in dual context modeling-based methods is enhanced by local-view and global-view simultaneously. We also note that the performance of all the models deteriorates as the session length increases. This decline can be attributed to the presence of a greater number of noisy clicks (Zhang, Lin, et al., 2022) and multiple user intents (Zhang et al., 2023) in longer sessions. Therefore, it is challenging for these models to precisely predict user behaviors under such complex circumstances.

5.8. Model efficiency

To address RQ8, we compare the efficiency of Coase with six representative baseline models: RepeatNet (Ren et al., 2019), TiSASRec (Li et al., 2020), CL4SRec (Xie et al., 2022), TAGNN (Yu et al., 2020), GCEGNN (Wang et al., 2020), and CMGNN (Wang, Gao, et al., 2023). All models are trained on a single Nvidia 3090 GPU, with the embedding dimension and batch size set to 100. For each epoch across the four datasets, we evaluate GPU memory consumption, training time, and inference time, as summarized in Table 12. The results suggest that Coase achieves a strong balance between space and time efficiency while maintaining competitive performance. Specifically, compared to dual-context modeling methods GCEGNN and CMGNN, Coase reduces GPU memory consumption by 93.67% and 93.68%, training time by 36.45% and 60.56%, and inference time by 7.20% and 13.57% on average. In addition, Coase requires less GPU memory, training time, and inference time than RepeatNet. While maintaining similar GPU memory consumption as TiSASRec, Coase achieves shorter inference times. It also demonstrates lower training times compared to CL4SRec and TAGNN. In summary, Coase offers a more effective trade-off between efficiency and performance than existing baselines, highlighting its potential for scalable and efficient deployment in real-world applications.

5.9. Visualization studies

In the first set of ablation studies (see Section 5.5.1), we identified that the global contextual feature extractor has the greatest impact on model performance, with SudokuFormer serving as a pivotal element within this module. To gain deeper insights into its effectiveness, we will illustrate the attention patterns of SudokuFormer through visualization studies. Specifically, the first set of visualizations shows the overall attention pattern of SudokuFormer, highlighting how it captures key interactions within sessions. The second set reveals how SudokuFormer operates differently compared to other models, offering insights into its unique ability to handle sequential and temporal information. Lastly, the third set demonstrates the significance of each component within

Table 12
Summary of efficiency results on four datasets.

Dataset	Method	GPU memory (MB)	Training time (s)	Inference time (s)
Yoochoose 1/64	RepeatNet	840	1213.55	98.11
	TiSASRec	599	100.27	52.61
	CL4SRec	611	1842.91	30.54
	TAGNN	3316	142.45	30.76
	GCEGNN	13 270	513.95	47.18
	CMGNN	13 270	685.92	49.23
	Coase (Ours)	654	298.80	40.46
Diginetica	RepeatNet	1470	3559.34	360.30
	TiSASRec	911	189.54	133.34
	CL4SRec	873	3532.30	72.26
	TAGNN	7456	586.11	99.04
	GCEGNN	13 286	958.99	107.86
	CMGNN	13 288	1323.16	111.21
	Coase (Ours)	944	562.67	110.17
Retailrocket	RepeatNet	1896	6660.26	389.77
	TiSASRec	1097	300.28	116.81
	CL4SRec	1053	5259.17	66.51
	TAGNN	10 124	1155.33	97.45
	GCEGNN	13 286	960.46	108.76
	CMGNN	12 304	2075.36	120.65
	Coase (Ours)	1102	773.86	103.84
Dressipi	RepeatNet	876	12 770.62	923.12
	TiSASRec	603	993.36	533.10
	CL4SRec	711	16 298.90	235.72
	TAGNN	3604	1477.09	338.51
	GCEGNN	13 272	4961.69	490.18
	CMGNN	15 480	8201.25	549.19
	Coase (Ours)	662	2819.71	430.47

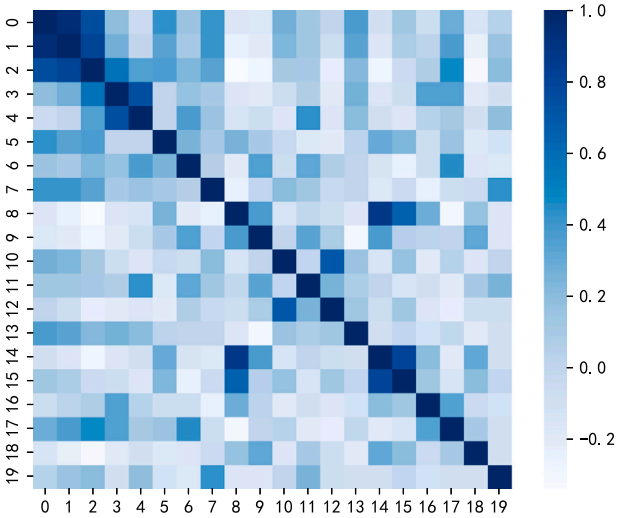


Fig. 12. Visualizations of the average self-attention weights for a random sampled batch of sessions learned by SudokuFormer, where dark regions indicate that the corresponding attention weights are promoted.

SudokuFormer, helping to explain the contribution of its design elements to model performance. Due to space limitations, all visualization studies are conducted on the Yoochoose 1/64 dataset.

Fig. 12 shows the averaged attention weights for a random sampled batch of sessions. It is observed that our SudokuFormer has a clear diagonal line effect for an item attending to itself. Moreover, we observe a scattered attention pattern where each target item has its own distinct sparse focus. It indicates that our SudokuFormer pays attention to some specific items on particular positions with special interests.

To visualize how SudokuFormer operates differently from other representative models, we present in Fig. 13 the normalized dot-product between two position encodings or time interval encodings from SudokuFormer, TiSASRec, SASRec, SGNN-HN, SLIME4Rec, and FEARec. It is observed that the position and time interval encodings of SudokuFormer exhibit similar yet distinct mixed patterns,

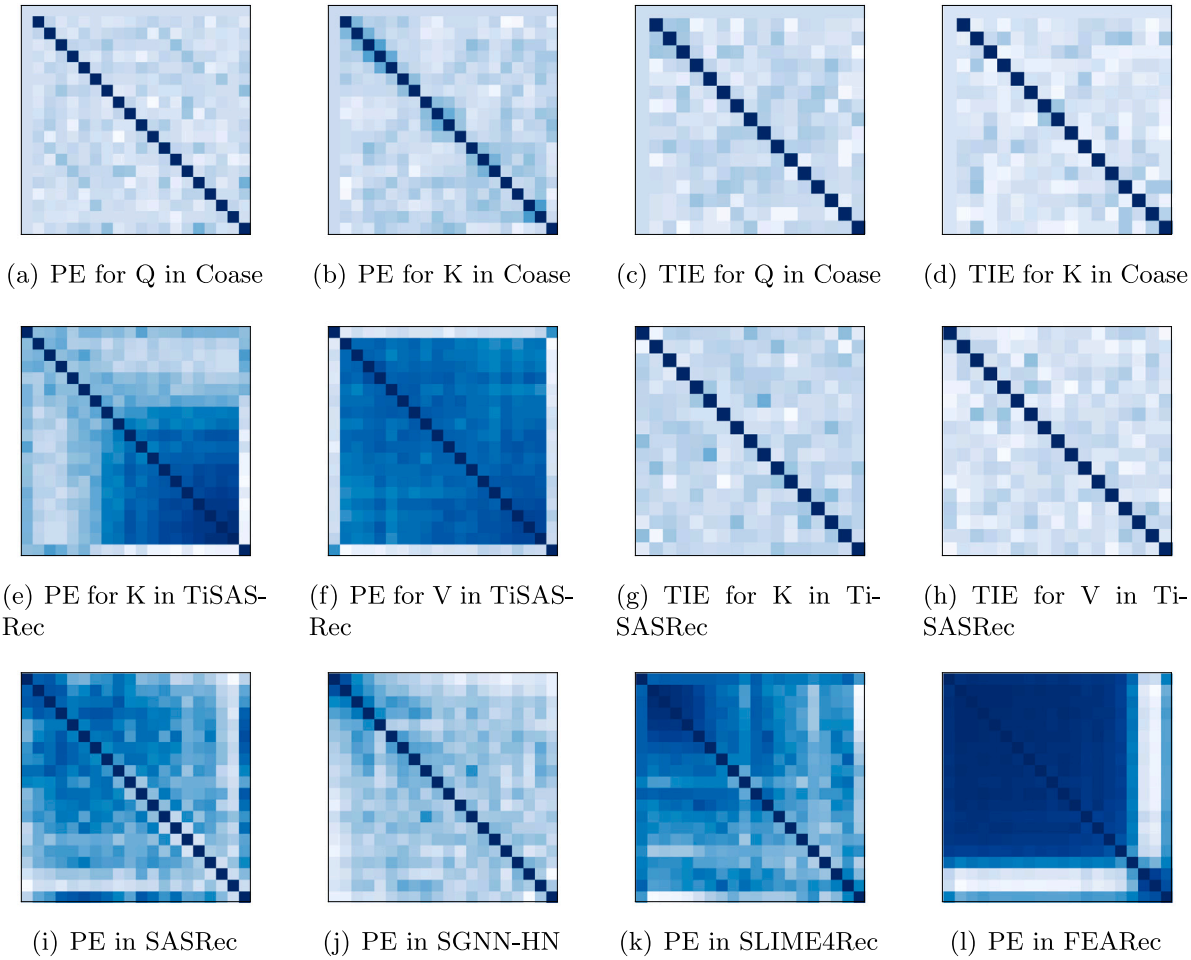


Fig. 13. Visualizations of normalized dot-product between any two encoding vectors in different models. PE and TIE denote the position encoding and the time interval encoding, respectively. Q, K, and V denote the key, query, and value, respectively. Darker means the two encoding vectors are closer.

as the disentangled learning effectively captures the hidden relationships among item encoding, position encoding, and time interval encoding. In contrast, TiSASRec's position and time interval encodings show significant differences, resulting in inconsistencies when capturing temporal information. Additionally, we observe varied patterns in the position encodings across models. For instance, SASRec, SGNN-HN, and SLIME4Rec tend to attend to all positions, with SASRec showing a more uniform pattern. By contrast, FEARec shows clear strips and blocks in its position encoding visualization, indicating that the last few positions are relatively independent of the others. While each model learns representative position encodings, most fail to fully capture the temporal information embedded in time intervals, which negatively impacts their performance.

To further investigate the effectiveness of SudokuFormer, we visualize the correlations in Eq. (19) for a random sampled batch of sessions, as shown in Fig. 14. We observe that different terms show its unique attention pattern. For example, the dark blocks with various size in the content-to-content term and the position-to-content term shows varying degrees of broad attention. Specifically, the first few items focus on more neighbor items in the content-to-content term. By contrast, the last few positions focus on more long-distance items in the position-to-content term. Furthermore, the sparse attention pattern represented by time interval-to-content term is more scattered, which helps the model capture more implicit information to improve its generalization. The phenomenon is consistent with our assumption that each term in the disentangled attention is not redundant.

6. Discussion and conclusion

This paper introduces Coase, a novel SBRS model that collaboratively integrates local and global context modeling. It utilizes a unified framework to combine both approaches, leveraging the collaborative effect within a multi-task learning setup. For local context modeling, Coase transforms session sequences into session star graphs and employs a Bi-Gated GSAN to learn item representations. For global context modeling, it applies position encoding and time interval encoding to capture various aspects of

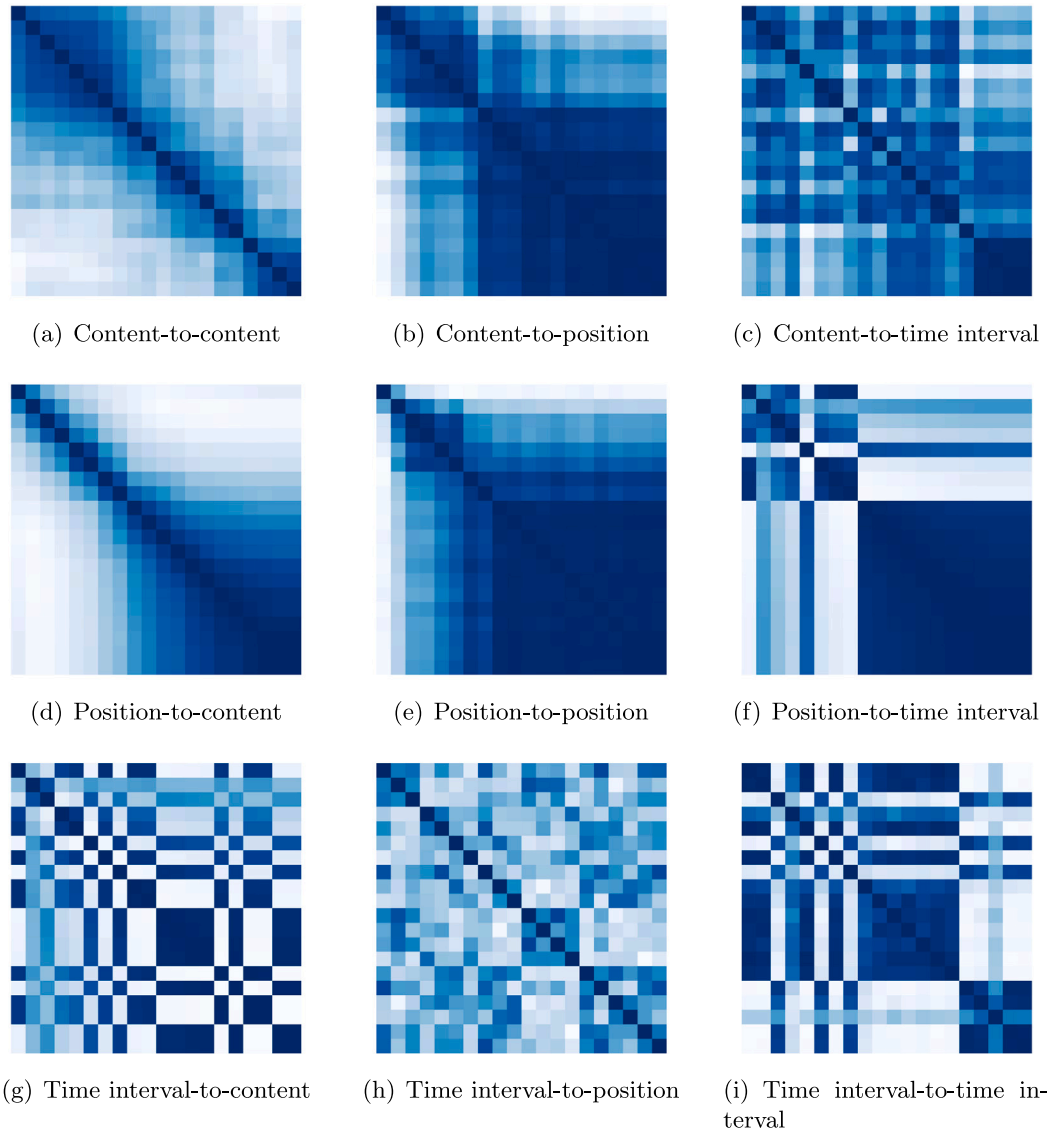


Fig. 14. Visualizations of the different attention terms on our SudokuFormer model for a random sampled batch of sessions. In each matrix, the (ith, jth) element is the correlation between ith item/position/time interval and jth item/position/time interval, where darker colors indicate higher correlations. We can find that different terms show its unique attention pattern.

temporal information. The SudokuFormer model is then used to update item features through a disentangled attention mechanism and a stable fusion method. Additionally, a triple attention mechanism is incorporated to fully capture user preferences, accounting for both short-term and long-term preferences, as well as the collaborative effect. Extensive experiments conducted on four real-world datasets demonstrate that Coase achieves state-of-the-art performance in session-based recommendation tasks. Ablation studies further validate the effectiveness of the framework, encoding methods, and components of SudokuFormer.

6.1. Theoretical implications

Our study makes significant methodological contributions for recommender systems. The proposed Coase is a novel SBRS model tailored for online platforms that delivers personalized services based on real-time user sessions. It achieves strong performance even without relying on external information. Compared with existing methods (Fu et al., 2025; Shin et al., 2024; Wang, Xie, et al., 2023), Coase offers several key advantages. First, Coase integrates graph learning and sequence learning paradigms to capture both local and global contextual features (Wan et al., 2024; Zhu et al., 2023) . This unified approach not only enhances the accuracy of item recommendations but also emphasizes the collaborative effects crucial for capturing detailed session-level user preferences. Second, for local context modeling (Pan, Cai, Chen, Chen, & Chen, 2022; Zhang, Xu, Wu, et al., 2024), Coase introduces Bi-Gated GSAN on session star graphs. This network highlights the intrinsic features of the central node, mitigating the risk of information

overload and ensuring more focused and meaningful message passing. Third, for global context modeling (Wang et al., 2022; Wang, Zhang, et al., 2024), we propose SudokuFormer, a novel technique that effectively analyzes the complex contextual relationships between items, their positions in the session sequence, and the timing of interactions. This results in more accurate and stable attention weights, leading to better recommendation quality.

6.2. Practical implications

Our comprehensive experiments in Section 5 validate Coase's effectiveness and superiority over 20 baseline models across four real-world datasets. These findings underscore Coase's robustness and adaptability in various recommendation environments, offering clear advantages for applications that demand high-accuracy, real-time suggestions. Additionally, the analysis of Coase's architecture highlights the significant impact of its core components on performance, with several important practical implications. First, by effectively integrating both local and global contextual features, Coase dynamically adapts to evolving user preferences. This capability is particularly valuable in real-time recommendation environments, such as e-commerce and streaming platforms, where user preferences can shift rapidly (Jannach et al., 2017). Practitioners can leverage this adaptability to deliver more relevant and timely recommendations, improving user engagement and satisfaction (Jannach & Jugovac, 2019). Second, instead of relying on external side information, Coase track dynamic fine-grained user preferences by considering various internal temporal information. In contrast to existing recommender systems (Chen et al., 2024; Wei et al., 2024; Zeng et al., 2025) that rely heavily on user profiles and long-term interaction histories, our method effectively balances recommendation precision and privacy protection. It achieves competitive performance in recommendations while utilizing only limited interaction history. Last but not least, our experiments demonstrate Coase's capability to effectively handle both short and long user sessions. This versatility suggests practical strategies for tailoring recommendation approaches based on session duration and user interaction type, thereby enhancing relevance in both exploratory and goal-oriented browsing sessions (Moe, 2003). Together, these results not only confirm Coase's effectiveness but also provide valuable insights for optimizing recommendation system design to meet diverse industry demands, from personalized marketing to customer retention and beyond.

6.3. Limitation and future work

While this research has yielded several notable findings and valuable contributions, we also acknowledge certain limitations. First, although we validated the effectiveness of the collaborative effect from the first set of ablation studies (see Section 5.5.1), its impact on recommendation performance was less pronounced compared to the complete removal of either the local or global contextual feature extractor. This prompts us to explore alternative paradigms for integrating both feature extractors in the future studies. Second, as shown by the experimental results in Table 11, Coase did not achieve optimal performance on longer sessions. This indicates that, in contexts where sessions are lengthy and training data is limited, Coase may not provide the best recommendation outcomes for users. Future research could address this by incorporating self-supervised learning techniques, such as knowledge distillation, to enhance performance in these scenarios.

CRediT authorship contribution statement

Weiyue Li: Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bowei Chen:** Writing – review & editing. **Ming Gao:** Writing – review & editing, Supervision, Resources. **Jingmin An:** Writing – review & editing. **Hao Dong:** Writing – review & editing, Visualization. **Cheng Chen:** Writing – review & editing. **Weiguo Fan:** Writing – review & editing. **Zhiguo Zhu:** Resources, Funding acquisition.

Acknowledgment

This research was supported and funded by the National Natural Science Foundation of China (No. 72172025, 72293563, 72101051, 72442025); the Humanities and Social Sciences Foundation of the Ministry of Education of China (No. 21YJAZH130); the Basic Scientific Research Project of Liaoning Provincial Department of Education (No. LJKMZ20221606, JYTZD2023050); the Natural Science Foundation of Liaoning Province (No. 2024-MS-175); the Liaoning Province Key Research and Development Project (No. 2024JH2/102400020); and the Dalian Scientific and Technological Talents Innovation Support Plan (No. 2022RG17). The authors would also like to thank Hongyu Wang of the Institute of Computing Technology, Chinese Academy of Sciences, for helpful discussions related to this work.

Appendix. Recommendation performance of examined models on each dataset

See Tables A.13–A.16.

Data availability

Data will be made available on request.

Table A.13
Performance of examined models on Yoochoose 1/64. The best-performing score in each case is highlighted in bold, while the second-best is underlined. A superscript * indicates that Coase significantly outperforms the second-best result, based on a paired t-test with a p-value < 0.05.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Traditional methods	5.04	2.69	3.25	7.95	3.09	4.21	10.07	3.26	4.77	12.69	3.40	5.38
Item-KNN		26.70	16.83	19.17	35.41	18.01	22.01	39.96	18.37	23.22	42.87	18.53	23.91
NextItNet	Local oriented methods	40.07	23.62	27.70	52.13	25.23	31.61	58.21	25.71	33.22	62.04	25.93	34.12
SRGNN		46.18	28.13	32.61	58.43	29.78	36.59	64.26	30.24	38.14	67.79	30.44	38.97
GCSAN		47.25	28.82	33.41	59.66	30.50	37.44	65.52	30.96	39.00	69.13	31.17	39.85
TAGNN		47.65	29.02	33.66	59.93	30.68	37.64	65.65	31.13	39.16	69.12	31.33	39.98
LESSR		46.86	28.48	33.05	59.18	30.14	37.05	64.94	30.60	38.58	68.49	30.80	39.41
GRU4Rec	Global oriented methods	45.72	27.93	32.36	57.87	29.57	36.30	63.69	30.03	37.84	67.22	30.23	38.68
NARM		46.14	28.20	32.66	58.53	29.87	36.68	64.43	30.34	38.25	67.85	30.53	39.06
STAMP		44.82	27.94	32.14	56.03	29.46	35.78	61.51	29.89	37.23	64.96	30.09	38.05
STAR		46.48	29.49	34.49	56.51	31.84	37.74	61.32	32.22	39.01	64.14	32.38	39.68
RepeatNet		45.42	27.23	31.75	57.35	28.84	35.62	63.01	29.29	37.12	66.45	29.48	37.94
CORE		44.79	26.60	31.12	57.05	28.25	35.10	63.19	28.74	36.73	66.80	28.95	37.59
SASRec		47.09	28.96	33.46	59.44	30.62	37.47	65.43	31.10	39.06	69.03	31.30	39.92
TiSASRec		47.83	29.17	33.81	60.11	30.83	37.80	66.09	31.30	39.39	69.60	31.50	40.22
AC-TSR		47.84	29.36	33.96	60.48	31.06	38.06	66.63	31.55	39.69	70.42	31.76	40.58
CL4SRec		46.08	28.41	32.81	58.08	30.02	36.70	64.21	30.51	38.32	67.84	30.71	39.18
DuoRec		46.33	28.70	33.08	58.08	30.27	36.89	63.97	30.74	38.45	67.65	30.95	39.32
TCP-SRec	Dual oriented methods	45.78	28.04	32.45	57.50	29.63	36.27	63.15	30.08	37.77	66.58	30.27	38.57
COTREC		38.61	28.66	31.15	44.87	29.49	33.17	48.32	29.76	34.07	50.89	29.91	34.68
SGNN-HN		46.44	28.57	33.01	58.54	30.21	36.94	64.32	30.67	38.48	67.88	30.87	39.32
GCEGNN		48.41	29.53	34.22	60.43	31.15	38.13	66.36	31.62	39.70	69.85	31.82	40.53
CMGNN		47.73	29.34	33.91	60.01	30.99	37.90	65.70	31.44	39.40	69.22	31.64	40.24
FEARec		46.62	28.79	33.23	58.34	30.37	37.04	64.24	30.84	38.60	67.85	31.05	39.46
SLIME4Rec		48.09	29.63	34.22	60.26	31.27	38.17	66.20	31.74	39.75	69.76	31.94	40.59
Coase (Ours)		**49.05***	**30.45***	**35.07***	**61.25***	**32.09***	**39.03***	**67.12***	**32.55***	**40.59***	**70.50***	**32.75***	**41.39***

Table A.14
Performance of examined models on Diginetica. The best-performing score in each case is highlighted in bold, while the second-best is underlined. A superscript * indicates that Coase significantly outperforms the second-best result, based on a paired t-test with a p-value < 0.05.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Traditional methods	0.52	0.23	0.30	0.95	0.29	0.44	1.29	0.32	0.53	1.59	0.33	0.60
Item-KNN		19.27	11.05	13.01	28.34	12.25	15.94	34.48	12.73	17.56	38.99	12.98	18.63
NextItNet	Local oriented methods	20.11	11.10	13.33	29.74	12.38	16.43	36.16	12.89	18.13	40.92	13.15	19.25
SRGNN		27.85	16.23	19.10	38.91	17.70	22.67	45.93	18.25	24.53	51.10	18.54	25.75
GCSAN		29.00	17.16	20.09	39.90	18.61	23.61	46.77	19.15	25.43	51.83	19.43	26.62
TAGNN		29.55	17.30	20.33	40.51	18.76	23.87	47.67	19.61	25.77	52.80	19.61	26.98
LESSR		28.42	16.55	19.49	39.73	18.05	23.13	46.70	18.60	24.98	51.80	18.89	26.19
GRU4Rec	Global oriented methods	26.98	15.57	18.39	37.98	17.03	21.94	45.00	17.58	23.80	50.14	17.87	25.01
NARM		27.68	15.99	18.88	38.88	17.48	22.50	45.93	18.04	24.37	51.04	18.33	25.57
STAMP		25.55	14.87	17.51	35.76	16.23	20.81	42.38	16.75	22.56	47.33	17.03	23.73
STAR		29.30	18.49	21.17	38.66	19.74	24.19	44.50	20.20	25.74	48.75	20.44	26.74
RepeatNet		29.35	18.09	20.90	37.48	19.18	23.53	42.10	19.54	24.75	45.39	19.73	25.53
CORE		30.65	18.27	21.34	41.39	19.70	24.81	48.21	20.23	26.61	53.02	20.51	27.75
SASRec		28.76	16.91	19.85	39.93	18.40	23.45	47.04	18.96	25.33	52.23	19.25	26.56
TiSASRec		28.42	16.59	19.52	39.70	18.09	23.16	46.92	18.66	25.07	52.26	18.96	26.33
AC-TSR		26.63	15.80	18.48	37.37	17.22	21.94	44.39	17.77	23.79	49.59	18.06	25.02
CL4SRec		26.15	15.49	18.13	36.42	16.85	21.44	43.30	17.39	23.26	48.38	17.68	24.46
DuoRec		27.68	16.32	19.13	38.40	17.75	22.59	45.50	18.30	24.47	50.70	18.60	25.70
TCP-SRec	Dual oriented methods	22.46	12.30	14.81	33.15	13.72	18.25	40.42	14.29	20.18	45.83	14.59	21.46
COTREC		24.23	17.02	18.83	27.62	17.48	19.93	29.41	17.62	20.41	30.66	17.69	20.70
SGNN-HN		30.31	17.65	20.78	41.42	19.13	24.38	48.39	19.68	26.22	53.44	19.97	27.42
GCEGNN		30.42	17.84	20.96	41.71	19.34	24.60	48.66	19.89	26.44	53.72	20.17	27.64
CMGNN		29.34	17.06	20.10	40.63	18.56	23.75	47.65	19.12	25.60	52.73	19.40	26.80
FEARec		28.36	16.78	19.64	39.18	18.21	23.13	46.20	18.76	24.99	51.38	19.05	26.21
SLIME4Rec		29.93	17.51	20.59	41.50	19.05	24.32	48.74	19.62	26.24	54.05	19.92	27.49
Coase (Ours)		**31.79***	**18.75***	**21.98***	**43.33***	**20.29***	**25.71***	**50.46***	**20.85***	**27.60***	**55.53***	**21.14***	**28.80***

Table A.15
Performance of examined models on Retailrocket. The best-performing score in each case is highlighted in bold, while the second-best is underlined. A superscript * indicates that Coase significantly outperforms the second-best result, based on a paired t-test with a p-value < 0.05.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Traditional methods	0.44	0.23	0.28	0.72	0.27	0.37	0.90	0.29	0.42	1.24	0.31	0.50
Item-KNN		9.75	6.36	7.08	13.35	6.83	8.22	15.53	7.00	8.80	17.05	7.09	9.15
NextItNet	Local oriented methods	45.36	33.61	36.54	52.45	34.56	38.84	56.27	34.86	39.85	58.79	35.00	40.45
SRGNN		54.65	40.90	44.35	61.56	41.84	46.60	65.05	42.11	47.52	67.40	42.25	48.07
GCSAN		55.56	42.09	45.47	61.92	42.94	47.53	65.21	43.20	48.40	67.44	43.33	48.93
TAGNN		56.72	42.52	46.08	63.35	43.42	48.24	66.58	43.67	49.10	68.72	43.79	49.60
LESSR		55.79	41.76	45.28	62.33	42.64	47.40	65.67	42.91	48.29	67.88	43.03	48.81
GRU4Rec	Global oriented methods	54.44	41.17	44.50	61.14	42.08	46.67	64.58	42.35	47.59	66.89	42.48	48.13
NARM		54.88	41.43	44.80	61.63	42.34	46.99	65.19	42.62	47.94	67.49	42.75	48.48
STAMP		49.02	38.70	41.27	55.57	39.58	43.40	59.24	39.87	44.37	61.77	40.01	44.97
STAR		48.73	39.13	41.53	54.26	39.87	43.32	57.37	40.12	44.15	59.35	40.23	44.61
RepeatNet		52.87	40.41	43.55	57.25	41.01	44.98	59.38	41.18	45.54	60.83	41.26	45.89
CORE		54.60	40.49	44.03	61.92	41.48	46.40	65.73	41.78	47.41	68.22	41.92	48.00
SASRec		53.18	41.11	44.12	60.45	42.08	46.48	64.37	42.40	47.52	67.02	42.54	48.15
TiSASRec		56.14	42.33	45.79	62.96	43.25	48.01	66.48	43.53	48.94	68.80	43.66	49.49
AC-TSR		52.33	40.65	43.57	59.50	41.61	45.89	63.34	41.92	46.91	66.05	42.07	47.55
CL4SRec		51.29	39.78	42.65	58.22	40.71	44.90	61.95	41.01	45.89	64.53	41.15	46.50
DuoRec		52.53	40.66	43.62	59.80	41.64	45.98	63.79	41.95	47.04	66.45	42.10	47.66
TCP-SRec	Dual oriented methods	44.04	34.72	37.04	49.95	35.51	38.96	53.20	35.77	39.82	55.42	35.89	40.34
COTREC		44.32	40.92	41.79	45.21	40.99	41.95	45.77	41.01	42.03	46.20	41.03	42.08
SGNN-HN		55.84	41.01	44.73	63.00	41.98	47.06	66.59	42.26	48.01	68.92	42.40	48.56
GCEGNN		56.43	42.02	45.64	63.26	42.94	47.85	66.74	43.22	48.78	69.07	43.35	49.33
CMGNN		55.46	41.14	44.73	62.40	42.07	46.98	65.92	42.35	47.91	68.30	42.48	48.47
FEARec		52.87	40.77	43.80	60.18	41.76	46.17	64.08	42.06	47.20	66.69	42.21	47.82
SLIME4Rec		55.71	42.17	45.56	62.97	43.14	47.91	66.76	43.44	48.92	69.28	43.59	49.51
Coase (Ours)		**56.90***	**42.91***	**46.42***	**63.73***	**43.83***	**48.63***	**67.16***	**44.10***	**49.54***	**69.47***	**44.23***	**50.09***

Table A.16
Performance of examined models on Dressipi. The best-performing score in each case is highlighted in bold, while the second-best is underlined. A superscript * indicates that Coase significantly outperforms the second-best result, based on a paired t-test with a p-value < 0.05.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Traditional methods	1.30	0.74	0.87	2.17	0.85	1.14	2.84	0.91	1.32	3.39	0.94	1.45
Item-KNN		13.30	7.94	9.06	19.35	8.76	11.03	23.38	9.09	12.11	26.40	9.26	12.83
NextItNet	Local oriented methods	19.68	11.67	13.66	27.11	12.66	16.05	31.91	13.04	17.32	35.47	13.24	18.17
SRGNN		22.19	13.33	15.52	29.93	14.36	18.02	34.81	14.74	19.31	38.34	14.94	20.15
GCSAN		23.05	13.74	16.05	31.06	14.81	18.64	36.11	15.21	19.97	39.77	15.41	20.84
TAGNN		23.35	13.72	16.10	31.60	14.82	18.77	36.69	15.22	20.11	40.38	15.42	20.99
LESSR		21.55	13.01	15.13	29.18	14.03	17.59	33.95	14.40	18.85	37.52	14.60	19.70
GRU4Rec	Global oriented methods	22.18	13.12	15.36	30.36	14.21	18.01	35.49	14.61	19.36	39.23	14.82	20.24
NARM		21.79	12.90	15.10	29.90	13.98	17.72	35.02	14.38	19.07	38.79	14.59	19.96
STAMP		20.91	12.78	14.80	27.98	13.73	17.08	32.52	14.08	18.28	35.87	14.27	19.07
STAR		19.89	12.48	14.32	26.16	13.31	16.34	30.20	13.63	17.41	33.24	13.80	18.13
RepeatNet		21.81	13.46	15.53	28.89	14.40	17.82	33.51	14.76	19.04	36.80	14.95	19.82
CORE		20.44	11.89	14.01	27.99	12.90	16.45	32.66	13.27	17.69	36.10	13.46	18.50
SASRec		21.58	12.96	15.09	29.41	14.00	17.62	34.47	14.40	18.96	38.21	14.61	19.84
TiSASRec		23.08	**13.99**	16.24	31.05	15.05	18.82	36.11	15.45	20.16	39.82	15.66	21.03
AC-TSR		20.93	12.95	14.93	28.25	13.92	17.29	33.03	14.30	18.56	17.29	14.50	19.40
CL4SRec		19.30	11.73	13.60	26.33	12.66	15.87	30.99	13.03	17.10	34.48	13.22	17.93
DuoRec		21.17	13.15	15.14	28.42	14.11	17.48	33.16	14.49	18.73	36.66	14.68	19.56
TCP-SRec	Dual oriented methods	18.33	10.55	12.47	25.43	11.50	14.77	29.97	11.86	15.97	33.36	12.05	16.77
COTREC		17.08	12.06	13.31	21.12	12.59	14.61	23.84	12.80	15.33	25.97	12.92	15.83
SGNN-HN		22.64	13.14	15.49	30.88	14.24	18.15	36.03	14.64	19.52	39.78	14.85	20.40
GCEGNN		23.38	13.82	16.19	31.34	14.89	18.77	36.30	15.28	20.08	39.92	15.48	20.93
CMGNN		**23.69**	13.93	16.35	31.78	15.01	18.97	36.78	15.41	20.29	40.48	15.62	21.17
FEARec		20.47	12.71	14.63	27.39	13.62	16.86	31.90	13.98	18.06	35.26	14.17	18.85
SLIME4Rec		22.40	13.61	15.79	30.18	14.64	18.30	35.13	15.03	19.61	38.79	15.24	20.47
Coase (Ours)		23.65	**13.99**	**16.38**	**32.03**	**15.11**	**19.09**	**37.23**	**15.51**	**20.46**	**41.03**	**15.73**	**21.36**

References

Chen, M., Burke, R. R., Hui, S. K., & Leykin, A. (2021). Understanding lateral and vertical biases in consumer attention: An in-store ambulatory eye-tracking study. *Journal of Marketing Research*, *58*(6), 1120–1141, URL http://dx.doi.org/10.1177/0022243721998375.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of machine learning research*: *119*, *International conference on machine learning* (pp. 1597–1607). URL http://proceedings.mlr.press/v119/chen20j.html.

Chen, P., Tsai, H., Bhojanapalli, S., Chung, H. W., Chang, Y., & Ferng, C. (2021). Demystifying the better performance of position encoding variants for transformer. *CoRR*, arXiv:2104.08698.

Chen, T., & Wong, R. C. (2020). Handling information loss of graph neural networks for session-based recommendation. In *ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1172–1180).

Chen, L., Zhu, G., Liang, W., Cao, J., & Chen, Y. (2024). Keywords-enhanced contrastive learning model for travel recommendation. *Information Processing & Management*, *61*(6), Article 103874.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Conference of the association for computational linguistics* (pp. 2978–2988).

Dang, Y., Yang, E., Guo, G., Jiang, L., Wang, X., Xu, X., Sun, Q., & Liu, H. (2023). Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *AAAI conference on artificial intelligence* (pp. 4225–4232).

Dang, Y., Yang, E., Guo, G., Jiang, L., Wang, X., Xu, X., Sun, Q., & Liu, H. (2024). TiCoSeRec: Augmenting data to uniform sequences by time intervals for effective recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *36*(6), 2686–2700.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Vleet, T. V., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., & Sampath, D. (2010). The YouTube video recommendation system. In *ACM conference on recommender systems* (pp. 293–296).

de Reuver, M., & Bouwman, H. (2015). Dealing with self-report bias in mobile internet acceptance and usage studies. *Information & Management*, *52*(3), 287–294.

Du, X., Yuan, H., Zhao, P., Fang, J., Liu, G., Liu, Y., Sheng, V. S., & Zhou, X. (2023). Contrastive enhanced slide filter mixer for sequential recommendation. In *IEEE international conference on data engineering* (pp. 2673–2685).

Du, X., Yuan, H., Zhao, P., Qu, J., Zhuang, F., Liu, G., Liu, Y., & Sheng, V. S. (2023). Frequency enhanced hybrid attention network for sequential recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 78–88).

Dufter, P., Schmitt, M., & Schütze, H. (2020). Increasing learning efficiency of self-attention networks through direct position interactions, learnable temperature, and convoluted attention. In *International conference on computational linguistics* (pp. 3630–3636).

Dufter, P., Schmitt, M., & Schütze, H. (2022). Position information in transformers: An overview. *Computational Linguistics*, *48*(3), 733–763.

Fang, H., Zhang, D., Shu, Y., & Guo, G. (2020). Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems*, *39*(1), 10:1–10:42.

Fisher, G., & Woolley, K. (2024). How consumers resolve conflict over branded products: Evidence from mouse cursor trajectories. *Journal of Marketing Research*, *61*(1), 165–184, URL http://dx.doi.org/10.1177/00222437231170838.

Fu, H., Qin, Z., Xue, W., & Ding, G. (2025). Fusing temporal and semantic dependencies for session-based recommendation. *Information Processing & Management*, *62*(1), Article 103896.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of machine learning research*: *70*, *International conference on machine learning* (pp. 1243–1252). URL http://proceedings.mlr.press/v70/gehring17a.html.

Gomez-Uribe, C. A., & Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, *6*(4), 13:1–13:19.

Gong, S., & Zhu, K. Q. (2022). Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1185–1195).

Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., & Zhang, Z. (2019). Star-transformer. In *Conference of the North American chapter of the association for computational linguistics* (pp. 1315–1325).

Guo, J., Yang, Y., Song, X., Zhang, Y., Wang, Y., Bai, J., & Zhang, Y. (2022). Learning multi-granularity consecutive user intent unit for session-based recommendation. In *ACM international conference on web search and data mining* (pp. 343–352).

Guo, J., Zhang, P., Li, C., Xie, X., Zhang, Y., & Kim, S. (2022). Evolutionary preference learning via graph nested GRU ODE for session-based recommendation. In *ACM international conference on information and knowledge management* (pp. 624–634).

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 639–648).

He, P., Liu, X., Gao, J., & Chen, W. (2021). Deberta: Decoding-enhanced Bert with disentangled attention. In *International conference on learning representations* (pp. 1–9). URL https://openreview.net/forum?id=XPZIaotutsD.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *International conference on learning representations* (pp. 1–10). URL http://arxiv.org/abs/1511.06939.

Hosanagar, K., Fleder, D. M., Lee, D., & Buja, A. (2014). Will the global village fracture into tribes? Recommender systems and their effects on consumer fragmentation. *Management Science*, *60*(4), 805–823.

Hou, Y., Hu, B., Zhang, Z., & Zhao, W. X. (2022). CORE: simple and effective session-based recommendation within consistent representation space. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1796–1801).

Huang, Z., Liang, D., Xu, P., & Xiang, B. (2020). Improve transformer models with better relative position embeddings. In *Findings of ACL: EMNLP 2020*, *Conference on empirical methods in natural language processing* (pp. 3327–3335).

Huang, L., Ma, Y., Liu, Y., Du, B. D., Wang, S., & Li, D. (2023). Position-enhanced and time-aware graph convolutional network for sequential recommendations. *ACM Transactions on Information Systems*, *41*(1), 6:1–6:32.

Hui, S. K., Fader, P. S., & Bradlow, E. T. (2009). Path data in marketing: An integrative framework and prospectus for model building. *Marketing Science*, *28*(2), 320–335.

Jannach, D., & Jugovac, M. (2019). Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems*, *10*(4), 16:1–16:23.

Jannach, D., Ludewig, M., & Lerche, L. (2017). Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction*, *27*(3–5), 351–392.

Jayakumar, S. M., Czarnecki, W. M., Menick, J., Schwarz, J., Rae, J. W., Osindero, S., Teh, Y. W., Harley, T., & Pascanu, R. (2020). Multiplicative interactions and where to find them. In *International conference on learning representations* (pp. 1–16). URL https://openreview.net/forum?id=rylnK6VtDH.

Jiang, J., Zhang, P., Luo, Y., Li, C., Kim, J. B., Zhang, K., Wang, S., Xie, X., & Kim, S. (2023). Adamct: Adaptive mixture of CNN-transformer for sequential recommendation. In *ACM international conference on information and knowledge management* (pp. 976–986).

Kang, W., & McAuley, J. J. (2018). Self-attentive sequential recommendation. In *IEEE international conference on data mining* (pp. 197–206).

Karimi, M., Jannach, D., & Jugovac, M. (2018). News recommender systems - survey and roads ahead. *Information Processing & Management*, *54*(6), 1203–1227.

Kim, Y., Awadallah, A. H., White, R. W., & Zitouni, I. (2014). Modeling dwell time to predict click-level satisfaction. In *ACM international conference on web search and data mining* (pp. 193–202).

Landia, N., Mcalister, R., North, D., Kalloori, S., Srivastava, A., & Ferwerda, B. (2022). RecSys challenge 2022 dataset: Dressipi 1M fashion sessions. In *Proceedings of the recommender systems challenge 2022* (pp. 1–3). Association for Computing Machinery.

Larsen, N. M., Sigurdsson, V., Breivik, J., & Orquin, J. L. (2020). The heterogeneity of shoppers' supermarket behaviors based on the use of carrying equipment. *Journal of Business Research*, *108*, 390–400, URL https://www.sciencedirect.com/science/article/pii/S0148296319307908.

Li, N., Ban, X., Ling, C., Gao, C., Hu, L., Jiang, P., Gai, K., Li, Y., & Liao, Q. (2024). Modeling user fatigue for sequential recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 996–1005).

Li, Y., Gao, C., Luo, H., Jin, D., & Li, Y. (2022). Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1997–2002).

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *ACM international conference on information and knowledge management* (pp. 1419–1428).

Li, J., Wang, Y., & McAuley, J. J. (2020). Time interval aware self-attention for sequential recommendation. In *ACM international conference on web search and data mining* (pp. 322–330).

Liu, C., Li, X., Cai, G., Dong, Z., Zhu, H., & Shang, L. (2021). Noninvasive self-attention for side information fusion in sequential recommendation. In *AAAI conference on artificial intelligence* (pp. 4249–4256).

Liu, X., Yu, H., Dhillon, I. S., & Hsieh, C. (2020). Learning to encode position for transformer with continuous dynamical model. *119*, In *International conference on machine learning* (pp. 6327–6335). URL http://proceedings.mlr.press/v119/liu20n.html.

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: short-term attention/memory priority model for session-based recommendation. In *ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1831–1839).

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International conference on learning representations* (pp. 1–8). URL https://openreview.net/forum?id=Bkg6RiCqY7.

Moe, W. W. (2003). Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. *Journal of Consumer Psychology*, *13*(1), 29–39, URL https://www.sciencedirect.com/science/article/pii/S1057740803701740.

Pan, Z., Cai, F., Chen, W., & Chen, H. (2022). Graph co-attentive session-based recommendation. *ACM Transactions on Information Systems*, *40*(4), 67:1–67:31.

Pan, Z., Cai, F., Chen, W., Chen, C., & Chen, H. (2022). Collaborative graph learning for session-based recommendation. *ACM Transactions on Information Systems*, *40*(4), 72:1–72:26.

Pan, Z., Cai, F., Chen, W., Chen, H., & de Rijke, M. (2020). Star graph neural networks for session-based recommendation. In *ACM international conference on information and knowledge management* (pp. 1195–1204).

Peintner, A., Mohammadi, A. R., & Zangerle, E. (2023). SPARE: shortest path global item relations for efficient session-based recommendation. In *ACM conference on recommender systems* (pp. 58–69).

Press, O., Smith, N. A., & Lewis, M. (2021). Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (pp. 5493–5505).

Qiu, R., Huang, Z., Chen, T., & Yin, H. (2022). Exploiting positional information for session-based recommendation. *ACM Transactions on Information Systems*, *40*(2), 35:1–35:24.

Qiu, R., Huang, Z., Li, J., & Yin, H. (2020). Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Transactions on Information Systems*, *38*(3), 22:1–22:23.

Qiu, R., Huang, Z., Yin, H., & Wang, Z. (2022). Contrastive learning for representation degeneration problem in sequential recommendation. In *ACM international conference on web search and data mining* (pp. 813–823).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*, 140:1–140:67, URL https://jmlr.org/papers/v21/20-074.html.

Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., & de Rijke, M. (2019). RepeatNet: A repeat aware neural recommendation machine for session-based recommendation. In *AAAI conference on artificial intelligence* (pp. 4806–4813).

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *International world wide web conference* (pp. 285–295).

Shalaby, W., Oh, S., Afsharinejad, A., Kumar, S., & Cui, X. (2022). M2TRec: Metadata-aware multi-task transformer for large-scale and cold-start free session-based recommendations. In *ACM conference on recommender systems* (pp. 573–578).

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. In *Conference of the North American chapter of the association for computational linguistics* (pp. 464–468).

Shin, Y., Choi, J., Wi, H., & Park, N. (2024). An attentive inductive bias for sequential recommendation beyond the self-attention. In *AAAI conference on artificial intelligence* (pp. 8984–8992).

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958, URL https://dl.acm.org/doi/10.5555/2627435.2670313.

Strycharz, J., Smit, E. G., Helberger, N., & van Noort, G. (2021). No to cookies: Empowering impact of technical and legal knowledge on rejecting tracking cookies. *Computers in Human Behavior*, *120*, Article 106750.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *ACM international conference on information and knowledge management* (pp. 1441–1450).

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. In *Workshop on deep learning for recommender systems* (pp. 17–22).

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *ACM international conference on web search and data mining* (pp. 565–573).

Tang, G., Zhu, X., Guo, J., & Dietze, S. (2022). Time enhanced graph neural networks for session-based recommendation. *Knowledge-Based Systems*, *251*, Article 109204.

Tian, C., Lin, Z., Bian, S., Wang, J., & Zhao, W. X. (2022). Temporal contrastive pre-training for sequential recommendation. In *ACM international conference on information & knowledge management* (pp. 1925–1934).

Tuan, T. X., & Phuong, T. M. (2017). 3D convolutional networks for session-based recommendation with content features. In *ACM conference on recommender systems* (pp. 138–146).

Ursu, R. M., Zhang, Q., & Honka, E. (2023). Search gaps and consumer fatigue. *Marketing Science*, *42*(1), 110–136.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Annual conference on neural information processing systems* (pp. 5998–6008). URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wan, Z., Liu, X., Wang, B., Qiu, J., Li, B., Guo, T., Chen, G., & Wang, Y. (2024). Spatio-temporal contrastive learning-enhanced GNNs for session-based recommendation. *ACM Transactions on Information Systems*, *42*(2), 58:1–58:26.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2022). A survey on session-based recommender systems. *ACM Computing Surveys*, *54*(7), 154:1–154:38.

Wang, T., Chen, C., Huang, J., & Huang, S. (2023). Modeling cross-session information with multi-interest graph neural networks for the next-item recommendation. *ACM Transactions on Knowledge Discovery from Data*, *17*(1), 1:1–1:28.

Wang, C., Deng, Z., Lai, J., & Yu, P. S. (2019). Serendipitous recommendation in E-commerce using innovator-based collaborative filtering. *IEEE Transactions on Cybernetics*, *49*(7), 2678–2692.

Wang, F., Gao, X., Chen, Z., & Lyu, L. (2023). Contrastive multi-level graph neural networks for session-based recommendation. *IEEE Transactions on Multimedia*, *25*, 9278–9289.

Wang, E., Jiang, Y., Xu, Y., Wang, L., & Yang, Y. (2022). Spatial-temporal interval aware sequential POI recommendation. In *IEEE international conference on data engineering* (pp. 2086–2098).

Wang, F., & Liu, H. (2021). Understanding the behaviour of contrastive loss. In *IEEE conference on computer vision and pattern recognition* (pp. 2495–2504). URL https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html.

Wang, H., Ma, S., Dong, L., Huang, S., Zhang, D., & Wei, F. (2024). DeepNet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(10), 6761–6774.

Wang, H., Ma, S., Huang, S., Dong, L., Wang, W., Peng, Z., Wu, Y., Bajaj, P., Singhal, S., Benhaim, A., Patra, B., Liu, Z., Chaudhary, V., Song, X., & Wei, F. (2023). Magneto: A foundation transformer. In *Proceedings of machine learning research*: *202*, *International conference on machine learning* (pp. 36077–36092). URL https://proceedings.mlr.press/v202/wang23u.html.

Wang, Z., Wei, W., Cong, G., Li, X., Mao, X., & Qiu, M. (2020). Global context enhanced graph neural networks for session-based recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 169–178).

Wang, J., Xie, H., Wang, F. L., Lee, L., & Wei, M. (2023). Jointly modeling intra- and inter-session dependencies with graph neural networks for session-based recommendations. *Information Processing & Management*, *60*(2), Article 103209.

Wang, H., Yan, S., Wu, C., Han, L., & Zhou, L. (2023). Cross-view temporal graph contrastive learning for session-based recommendation. *Knowledge-Based Systems*, *264*, Article 110304.

Wang, H., Zeng, Y., Chen, J., Han, N., & Chen, H. (2023). Interval-enhanced graph transformer solution for session-based recommendation. *Expert Systems with Applications*, *213*(Part), Article 118970.

Wang, D., Zhang, X., Yin, Y., Yu, D., Xu, G., & Deng, S. (2024). Multi-view enhanced graph attention network for session-based music recommendation. *ACM Transactions on Information Systems*, *42*(1), 16:1–16:30.

Wei, Y., Xu, Y., Zhu, L., Ma, J., & Huang, J. (2024). FUMMER: a fine-grained self-supervised momentum distillation framework for multimodal recommendation. *Information Processing & Management*, *61*(5), Article 103776.

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2023). Graph neural networks in recommender systems: A survey. *ACM Computing Surveys*, *55*(5), 97:1–97:37.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In *AAAI conference on artificial intelligence* (pp. 346–353).

Wu, C., Wu, F., & Huang, Y. (2021). DA-transformer: Distance-aware transformer. In *Conference of the North American chapter of the association for computational linguistics* (pp. 2059–2068).

Xia, X., Yin, H., Yu, J., Shao, Y., & Cui, L. (2021). Self-supervised graph co-training for session-based recommendation. In *ACM international conference on information and knowledge management* (pp. 2180–2190).

Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., & Zhang, X. (2021). Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI conference on artificial intelligence* (pp. 4503–4511).

Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B., & Cui, B. (2022). Contrastive learning for sequential recommendation. In *IEEE international conference on data engineering* (pp. 1259–1273).

Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F., Fang, J., & Zhou, X. (2019). Graph contextualized self-attention network for session-based recommendation. In *International joint conference on artificial intelligence* (pp. 3940–3946).

Yang, Y., Zhang, J., Wang, Y., Miao, Z., & Tong, Y. (2023). Multiple connectivity views for session-based recommendation. In *ACM conference on recommender systems* (pp. 1000–1006).

Yao, Z., Chen, X., Wang, S., Dai, Q., Li, Y., Zhu, T., & Long, M. (2024). Recommender transformers with behavior pathways. In *ACM on web conference* (pp. 3643–3654).

Ye, W., Wang, S., Chen, X., Wang, X., Qin, Z., & Yin, D. (2020). Time matters: Sequential recommendation with complex temporal information. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1459–1468).

Yeganegi, R., Haratizadeh, S., & Ebrahimi, M. (2024). STAR: a session-based time-aware recommender system. *Neurocomputing*, *573*, Article 127104.

Yin, Z., Han, K., Wang, P., & Hu, H. (2023). Multi global information assisted streaming session-based recommendation system. *IEEE Transactions on Knowledge and Data Engineering*, *35*(8), 8245–8256.

Yin, Z., Han, K., Wang, P., & Zhu, X. (2024). H3GNN: hybrid hierarchical HyperGraph neural network for personalized session-based recommendation. *ACM Transactions on Information Systems*, *42*(3), 63:1–63:30.

Yu, B., Li, X., Fang, J., Tai, C., Cheng, W., & Xu, J. (2023). Memory-augmented meta-learning framework for session-based target behavior recommendation. *World Wide Web*, *26*(1), 233–251.

Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., & Tan, T. (2020). TAGNN: target attentive graph neural networks for session-based recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1921–1924).

Yuan, F., He, X., Jiang, H., Guo, G., Xiong, J., Xu, Z., & Xiong, Y. (2020). Future data helps training: Modeling future contexts for session-based recommendation. In *ACM on web conference* (pp. 303–313).

Yuan, J., Ji, W., Zhang, D., Pan, J., & Wang, X. (2022). Micro-behavior encoding for session-based recommendation. In *IEEE international conference on data engineering* (pp. 2886–2899).

Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J. M., & He, X. (2019). A simple convolutional generative network for next item recommendation. In *ACM international conference on web search and data mining* (pp. 582–590).

Zeng, J., Tao, H., Tang, H., Wen, J., & Gao, M. (2025). Global and local hypergraph learning method with semantic enhancement for POI recommendation. *Information Processing & Management*, *62*(1), Article 103868.

Zhang, Q., Cao, L., Shi, C., & Niu, Z. (2022). Neural time-aware sequential recommendation by jointly modeling preference dynamics and explicit feature couplings. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(10), 5125–5137.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B., & Liu, T. (2014). Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI conference on artificial intelligence* (pp. 1369–1375).

Zhang, P., Guo, J., Li, C., Xie, Y., Kim, J., Zhang, Y., Xie, X., Wang, H., & Kim, S. (2023). Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *ACM international conference on web search and data mining* (pp. 168–176).

Zhang, X., Lin, H., Xu, B., Li, C., Lin, Y., Liu, H., & Ma, F. (2022). Dynamic intent-aware iterative denoising network for session-based recommendation. *Information Processing & Management*, *59*(3), Article 102936.

Zhang, X., Xu, B., Ma, F., Li, C., Yang, L., & Lin, H. (2024). Beyond co-occurrence: Multi-modal session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *36*(4), 1450–1462.

Zhang, X., Xu, B., Wu, Y., Zhong, Y., Lin, H., & Ma, F. (2024). FineRec: Exploring fine-grained sequential recommendation. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 1599–1608).

Zhao, W. X., Hou, Y., Pan, X., Yang, C., Zhang, Z., Lin, Z., Zhang, J., Bian, S., Tang, J., Sun, W., Chen, Y., Xu, L., Zhang, G., Tian, Z., Tian, C., Mu, S., Fan, X., Chen, X., & Wen, J. (2022). RecBole 2.0: Towards a more up-to-date recommendation library. In *ACM international conference on information and knowledge management* (pp. 4722–4726).

Zhao, W. X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C., Min, Y., Feng, Z., Fan, X., Chen, X., Wang, P., Ji, W., Li, Y., Wang, X., & Wen, J. (2021). RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *ACM international conference on information and knowledge management* (pp. 4653–4664).

Zhou, P., Ye, Q., Xie, Y., Gao, J., Wang, S., Kim, J. B., You, C., & Kim, S. (2023). Attention calibration for transformer-based sequential recommendation. In *ACM international conference on information and knowledge management* (pp. 3595–3605).

Zhu, T., Sun, L., & Chen, G. (2023). Graph-based embedding smoothing for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 496–508.