A Flash Attention Transformer for Multi-Behaviour Recommendation

Tendai Mukande
SFI Centre for Research Training in Machine Learning
Dublin City University
Dublin, Ireland
tendai.mukande2@mail.dcu.ie

Esraa Ali
SFI Research Centre for AI-Driven Digital Content Technology
Dublin City University
Dublin, Ireland
esraa.ali@adaptcentre.ie

Annalina Caputo
School of Computing
Dublin City University
Dublin, Ireland
annalina.caputo@dcu.ie

Ruihai Dong
School of Computer Science
University College Dublin
Dublin, Ireland
ruihai.dong@ucd.ie

Noel E. O'Connor
Insight SFI Research Centre for Data Analytics
Dublin City University
Dublin, Ireland
noel.oconnor@dcu.ie

ABSTRACT

Recently, modelling heterogeneous interactions in recommender systems has attracted research interest. Real-world scenarios involve sequential multi-type user-item interactions such as "*view*", "*add-to-favourites*", "*add-to-cart*" and "*purchase*". Graph Neural Network (GNN) methods have been widely adopted in Representation Learning of similar sequential user-item interactions. Promising results have been achieved by the integration of GNNs and transformers for self-attention. However, GNN based methods suffer from limited capability in handling global user-item interaction dependencies, particularly for long sequences. Moreover, these models require high computational cost of transformers, due to the quadratic memory and time complexity with respect to sequence length. This results in memory bottlenecks and slow training especially in computational resource-constrained environments. To address these challenges, we propose the FATH model which employs Flash Attention mechanism to reduce the high-bandwidth memory usage over higher-order user-item interaction sequences. Experimental results show that our model improves the training speed and reduces the memory usage with better recommendation performance in comparison with the state-of the art baselines.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Flash Attention; Transformer; Multi-behaviour Recommendation; Graph Neural Networks

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615206

ACM Reference Format:
Tendai Mukande, Esraa Ali, Annalina Caputo, Ruihai Dong, and Noel E. O'Connor. 2023. A Flash Attention Transformer for Multi-Behaviour Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615206

1 INTRODUCTION

Recommender systems (RSs) have been essential in alleviating information overload to users by generating personalized suggestions. Challenges in modelling real-world scenarios for personalized RSs using deep neural networks have indeed garnered substantial research attention [36]. These challenges stem from the complexity and nuances of real-world data and user behavior. In real-world situations with restricted computational resources, a key concern arises: can recommendation performance be enhanced with reduced memory usage costs? Reducing the computational cost does not only reduce the monetary bill, for instance in the cloud, but also has a positive impact on the environment by overall reduction of the carbon footprint [20, 30]. This study focuses on the creating GNNs that tackle these challenges.

Self-attention GNN models have been proposed by recent works [25] to address the limitations of traditional methods, such as Collaborative Filtering and Matrix Factorization [41], to model real-world user-item relations for RSs. In these studies, GNNs leverage complex user and item features to generate recommendations using link prediction [1, 36]. Most existing research, however, has focused on single type user behaviour modelling whereas in the real practical scenarios user behaviour is multi-type [1, 38, 45]. In addition, most of these methods are limited to pairwise interactions, hence they do not efficiently represent multi-order relations [40] which is inconsistent with real-world scenarios wherein both pairwise and higher order relations may be encountered in the same sequence [5]. Another issue with state-of-the-art GNN models is high computational cost due to the use of transformers for self-attention, and have quadratic memory and time requirements with respect to sequence length [4, 26]. As a result of this, the models have a memory bottleneck and their training process is slower [23].

Several multi-behaviour recommendation models approach have been proposed [38–40, 45], in which collaborative behaviour such as "*view*","*add-to-favourites*", "*add-to-cart*" is used to predict the target "*purchase*" behaviour [36, 44]. While the recommendation performance has been shown to improve using these methods, the high computational cost issue, particularly over long user-item interaction sequences remains an active research direction [29]. Another issue with GNN methods is the limitation of the message-passing scheme, whereby node features are propagated according to the input neighborhood information, and they are unable to capture dependencies between nodes nodes with a longer distance than their message-passing steps [7].

To address the highlighted challenges, we propose the **Hypergraph Flash Attention Transformer for Multi-Behaviour Recommendation (FATH)**, which models the multi-behaviour data as high-order Multi-Layer Perceptron (MLP) linear layers, to capture more intricate patterns beyond the capabilities of the traditional GNN message-passing scheme. In this way, the model is able to learn both the local neighborhood user-item interactions as well as global dependencies such as node-to-edge or edge-to-node and interactions and between nodes which have no direct connection [15]. We further adopt Flash Attention to minimize the number High Bandwidth Memory (HBM) accesses during the attention computations [2]. Our model uses fewer memory and computational resources compared to the conventional Vanilla Transformer, achieved by minimizing data transfers between HBM, SRAM, and back.

The main hypothesis is that the efficient modelling of both the local and the global interactions enhances the representational capacity of the model, ultimately boosting recommendation performance. Additionally, earlier research has demonstrated that reducing HBM accesses results in quicker convergence and decreased memory utilization [2].

Our key **contributions** can be outlined as follows:

We introduce a novel GNN approach that enhances the recommendation performance and overcome the constraints of message-passing GNNs in capturing extensive dependencies. Our approach integrates a higher-order MLP-layer hypergraph model [15].

The proposed approach also optimizes memory and time efficiency within the model. This is achieved by incorporating blockwise flash attention mechanism [2], thereby reducing GPU HBM read and write operations.

In our experiments, we apply the proposed FATH model to the IJCAI dataset widely used in literature for recommendation tasks. Experimental results demonstrate that our model improves the recommendation performance and reduces the computational cost improvement compared to the state-of-the-art baselines.

2 RELATED WORK

Recent studies have used GNN methods to model heterogeneous data in RSs [33]. Graph Convolution Networks (GCNs) [28, 35] have shown promising results in various RS and Information Retrieval tasks [5, 43]. A limitation observed in GCNs is the fixed topology post-training. Efforts to mitigate this concern have resulted in the introduction of hybrid GCN-based self-attention approaches, which involve integrating transformers to overcome this problem [31]. Transformer-based recommendation models have been employed to manage dynamic scenarios, such as those encountered in session-based and sequential recommendation tasks. This includes plain transformer-based models [13, 27] as well as hybrid GNNs [42]. The main drawback of this family of methods is the high computational cost. Several methods been proposed to reduce the high computational cost of transformers [9, 16–18, 21, 24].

An alternative strategy for enhancing RS performance is through the adoption of multi-behavior recommendation models [1, 39]. These models leverage collaborative signals such as views, clicks, and *add-to-favorites* to improve predictions for the target behavior, which might involve rating or purchasing. While aligning well with real-world scenarios, research in this domain encounters constraints rooted in the foundational models, such as memory bottlenecks within self-attention modules and the challenges posed by higher-order user connections in GNN-based models [3, 14, 22]. In our work, we further advance the multi-behavior approach by employing a hypergraph MLP-based representation model coupled with a sub-quadratic attention module, thereby enhancing recommendation performance while concurrently reducing computational demands.

3 METHODOLOGY

3.1 Approach Overview

Our proposed methodology leverages the hypergraph neural network model to depict higher-order interaction sequences between users and items. This framework enables the generation of embeddings that effectively capture multi-type user-item interactions. To overcome the computational cost bottleneck, our approach incorporates the Flash Attention mechanism introduced by Dao et al. [2]. This mechanism reduces the frequency of memory reads/writes to memory bandwidth.

The underlying assumption is that by combining both of these techniques, we can achieve a reduction in the overall time required for attention computations as well as a decrease in GPU RAM utilization. Consequently, our recommendation approach gains the capability to process more extensive and multi-context user-item interaction sequences at an accelerated pace and with reduced computational expenditure. This integration ultimately translates into an improved recommendation performance.

Problem Formulation. In the e-commerce domain, we denote the user as u_i and an item as v_j. The set of users is $u_i \in \mathcal{U}$ where $|\mathcal{U}| = I$, I is total number of users. For user u_i, we denote the interaction sequence as $S_i = \left[\left(v_{i,1}, b_{i,1}\right), \ldots, \left(v_{i,j}, b_{i,j}\right), \ldots, \left(v_{i,J}, b_{i,J}\right)\right]$ where J is the length item sequence length. We also denote $b_{i,j}$ as the behavior interaction type between user u_i and item v_j in S_i. Instances of behavior types include *view*, *add-to-cart*, *add-to-favorite*, and *purchase*. When our focus is on forecasting the *purchase* behavior, the supplementary collaborative behaviors such as *view*, *add-to-favorites*, and *add-to-cart* contribute supplementary contextual insights into user preferences. In sequential recommendation, the input is the interaction sequence S_i of user u_i. The output is the function which estimates the probability of user u_i interaction with item v_{J+1} for the target behavior at time $(J + 1)$.

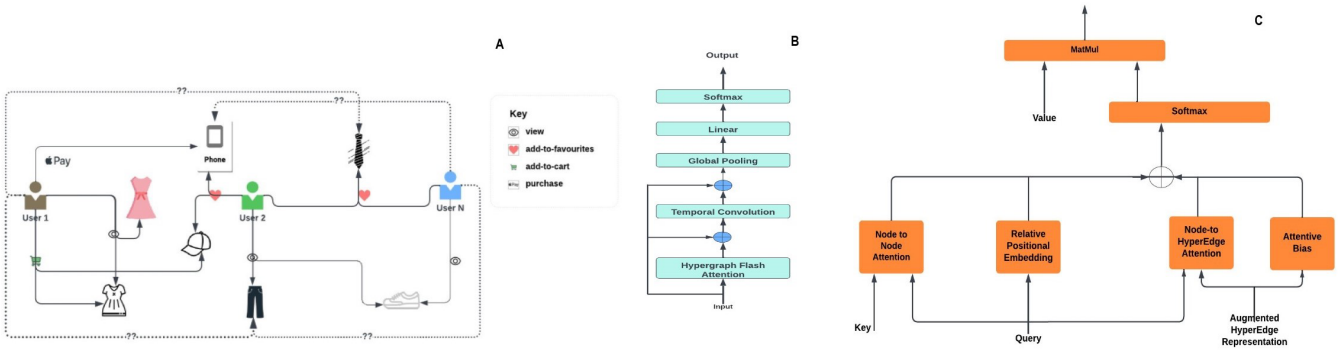

Figure 1: (A) Most real-world e-commerce recommendation scenarios consist of multi-order user-item interactions. (B) Recommendation model workflow. (C) The Hypergraph-Flash Attention module.

3.2 Memory Overview

The GPU's memory structure comprises High Bandwidth Memory (HBM), serving as the principal memory, and Static Random Access Memory (SRAM), functioning as the rapid cache memory [10]. In terms of characteristics, SRAM is both quicker and more compact than HBM. Throughout the execution of the model, GPU operations (kernels) retrieve input data from HBM, perform computations in the SRAM, and subsequently store the output back in HBM [12]. Kernels are classified as either compute-bound or memory-bound depending on the number of arithmetic operations per memory access [19]. In standard self-attention, intermediate attention outputs need to be written to HBM for the forward and backward passes, which results in more accesses between the HBM and SRAM, resulting in high memory and time complexity [21].

3.3 Hypergraph Model

The proposed hypergraph model includes high-order MLP-Layers to represent the multi-typed user-item interactions. The users/items are represented by the nodes whereas hyperedges connect users to items they have previously interacted with. To make personalized recommendations to a user, the *Query* vector represents their preferences based on a weighted combination of their past interactions. The *Key* vector captures important item features, and the *Value* vectors contain the information used to make recommendations.

Hypergraph Flash-Attention. For each user's *Query* vector , similarity scores are computed with *Key* vectors of items and attributes to represent their interest in those items. These scores are used as attention weights to aggregate the corresponding *Value* vectors of items and attributes, which captures the items alignment with the user preferences. The highest ranked items then recommended to the user. To jointly encode the global and local user-item interaction dependencies, the attention output is computed by a summation of the attention scores of the *node-to-node* attention, *the node-to-hyperedge* attention between the *Query* vectors and the corresponding hyperedge of the *Key* vectors, information sequence with the *Relative Positional Embeddings (RPE)*, and the augmented hyperedge representations [47].

During the attention computations, the objective is to enable faster training and use less memory. This is achieved by splitting the input sequence data into blocks and then use the tiling and recomputation techniques to reduce the number of read/write operations between the GPU SRAM and HBM.

The input data is split into blocks of query, key, and value from GPU HBM to SRAM and several forward passes are made over the input blocks. The tiling technique is performed by incrementals softmax reduction over the blocks instead of the whole sequence. Instead of reading the intermediate attention matrix from the HBM as in standard attention, the softmax normalization factor from the forward pass is stored and used in the SRAM for the backward pass attention computation without the storing and reading the intermediate matrix from the HBM [2].

Since the attention matrix is processed in tiles, with only the immediate softmax and exponential weighted sums being stored on-chip, the GPU memory usage is reduced [29].

4 EXPERIMENTS

In conducting our experiments, we aim to answer the **research questions** outlined below:

- **RQ1:** Does the FATH approach outperform existing state of the art baselines?
- **RQ2:** Does the Flash Attention mechanism reduce the model computational cost in terms of memory requirements and runtime for model training and evaluation?

Experimental Setup: To recommend an item to a user for potential purchase preference, it is necessary to take into account the previous multi-behavior interactions by users for the particular item. For evaluation, we use the leave-one-out strategy and follow the settings described in [42] and [27]. We obtain the test samples from the last purchase for each user and the validation samples from the previous purchases.

Dataset: For our experiments, we use the **IJCAI** real-world e-commerce dataset as pre-processed by Yang et al. [42]. This dataset was released for the IJCAI 2015 Contest and it contains *purchase, add-to favorites, add-to-cart* and *view* user-item interactions. The IJCAI dataset contains (200, 000) users, (808, 354) items, and (13, 072, 940) interactions.

Metrics: In order to evaluate the recommendation performance, the Hit Ratio (HR) , Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR) [34, 37]. Higher HR, NDCG and

Model	HR@5	NDCG@5	HR@10	NDCG@10	MRR
GRU4Rec	0.142	0.104	0.205	0.118	0.115
SASRec	0.144	0.109	0.190	0.122	0.120
BERT4Rec	0.298	0.218	0.408	0.256	0.228
HyperRec	0.138	0.105	0.232	0.140	0.130
NMTR	0.112	0.078	0.238	0.146	0.134
MB-GCN	0.216	0.144	0.337	0.184	0.178
MBHT	0.342	0.265	0.434	0.296	0.274
FATH	**0.413***	**0.286***	**0.465***	**0.312***	**0.297***

Table 1: Performance Comparison: Bold * indicates the performance improvement by our approach over the best performing baseline at 0.05 significance with paired t-test.

MRR values indicate better performance. For memory usage assessment we report the RAM memory usage during training and inference. Lower memory requirement indicates better efficiency. Finally, for the runtime comparison we compare the training and evaluation time (per epoch) of our model against other baselines. A lower runtime indicates a faster model.

Model Learning Setup: Our experiments are conducted using the Pytorch RecBole framework [46]. In order to train our model, the block size used is 32 and the sequence length is 4096. For fair comparison with other baselines, we follow the settings used by authors in [42] and [27] for sequential recommendation. The GPU hardware used is NVIDIA Tesla A100.

Baselines: We compare our approach to three groups of state of the art recommendation methods: 1) Sequential Methods, 2) Graph-based Methods, 3) Multi-behaviour Methods. For the first group we include **GRU4Rec**: A recurrent neural network model that uses a gated recurrent unit encoder to learn dynamic user-item interactions [8]. We also include **SASRec**: A Self-attention model for sequential recommendation [13]. For the second group we include **NMTR**, a relationship cascading model the multi-type behaviour dependencies [6]. In addition, **BERT4Rec** uses a self-attentive bi-directional model for sequence modeling [27]. We include **HyperRec**: A hypergraph representation learning model for dynamic user-item interactions [32]. For the multi-behavior methods group we include **MB-GCN**, a graph convolutional network based embedding model for user behaviour modelling [11], and **MBHT**, A Multi-Behavior Hypergraph-enhanced Transformer model for learning cross-type behavior dependencies [42].

5 RESULTS AND DISCUSSION

Performance Evaluation (RQ1): From results in Table 1, multi-behavior recommendation models MB-GCN and NMTR achieve better recommendation performance compared to SASRec, HyperRec and GRU4Rec. This indicates the effectiveness of incorporating multi-type behavior context information into the representation learning process. With a hypergraph representation learning which captures long-range item dependencies across behavior types, the MBHT model has better recommendation performance than the other multi-behavior recommendation approaches. Our FATH method outperforms the MBHT model, which can be attributed to the difference in the modelling of the context information , particularly the local and global dependencies of the user and item interactions, shows the effectiveness of our model compared to the GNN message passing-scheme used in the MBHT model. Efficient attention also contributes to the improved recommendation performance, as shown by the superior performance of the BERT4Rec.

Model Variant	NDCG@5
FATH	**0.286***
FATH without Joint-to-hyperedge attention	0.262
FATH without Relative Positional Embedding	0.251
FATH without Hyperedge Attentive Bias	0.236
FATH without Node to Node attention	0.248

Table 2: Ablation Study results showing the effect of key modules on the model performance.

Ablation Study: We study the effect of key modules of our model in our model. Experimental results conducted by removing the components indicated in Table 2 show a reduced performance of the model in terms of the NDCG metric, which shows their effectiveness in the model.

Model	Runtime (sec/epoch)	GPU Memory (GB)	NDCG@10
SASRec	1,784	12.36	0.122
MBHT	1,588	11.28	0.296
BERT4Rec	1,968	13.68	0.256
FATH with Vanilla	2,486	14.75	**0.313***
FATH	**1,458**	**10.42***	0.312

Table 3: Efficiency Evaluation Results. The lowest memory usage and runtime are indicated in bold.

Efficiency Evaluation (RQ2): We examine FATH model's computational cost in comparison to self-attention models SASRec, BERT4Rec, and MBHT (with low-rank attention). Memory and training efficiency are compared at sequence length 4096, batch size 64, 16 heads, and head dimension 64. Table 3 reveals our model's superiority in time and memory requisites. This stems largely from the Flash Attention mechanism, curtailing reads/writes to HBM, leading to enhanced speed and reduced memory usage versus transformer-based models. Notably, utilization of Flash Attention markedly alters model memory and runtime, diverging from the standard Vanilla Transformer approach.

6 CONCLUSION

In this paper, we proposed the FATH model which uses Flash Attention on an MLP-based hypergraph model for multi-behaviour product recommendation. Experimental results on the IJCAI real-world dataset demonstrate performance improvement with less memory usage and faster runtime by our model compared to the state-of-the-art baselines. In future, we plan to study our model performance on longer sequences as well as other GPU architectures.

ACKNOWLEDGMENTS

This publication has emanated from research supported by Science Foundation Ireland under Grant number 18/CRT/6183.

REFERENCES

[1] Huihui Chai, Xiumei Wei, Haoxiang Ma, and Xuesong Jiang. 2022. Knowledge-Enhanced Graph Transformer Network for Multi-Behavior and Item-Knowledge Session-based Recommendation. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 3421–3426.

[2] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* 35 (2022), 16344–16359.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[4] Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556* (2019).

[5] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3558–3565.

[6] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 1554–1557.

[7] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. 2020. Implicit graph neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 11984–11995.

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[9] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. 2022. Transformer quality in linear time. In *International Conference on Machine Learning*. PMLR, 9099–9117.

[10] Zhe Jia, Blake Tillman, Marco Maggioni, and Daniele Paolo Scarpazza. 2019. Dissecting the graphcore ipu architecture via microbenchmarking. *arXiv preprint arXiv:1912.03413* (2019).

[11] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.

[12] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*. 1–12.

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[14] Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595* (2020).

[15] Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. 2021. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems* 34 (2021), 28016–28028.

[16] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).

[17] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. *Advances in Neural Information Processing Systems* 32 (2019).

[18] Hanxiao Liu, Andy Brock, Karen Simonyan, and Quoc Le. 2020. Evolving normalization-activation layers. *Advances in Neural Information Processing Systems* 33 (2020), 13539–13550.

[19] Douglass Stott Parker. 1995. *Random butterfly transformations with applications in computational linear algebra.* UCLA Computer Science Department.

[20] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350* (2021).

[21] Markus N Rabe and Charles Staats. 2021. Self-attention Does Not Need $O(n^2)$ Memory. *arXiv preprint arXiv:2112.05682* (2021).

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

[23] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics* 9 (2021), 53–68.

[24] Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* (2020).

[25] Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey, and Prasenjit Choudhury. 2021. Recommender systems: an overview, research trends, and future directions. *International Journal of Business and Systems Research* 15, 1 (2021), 14–52.

[26] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864* (2021).

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[28] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[29] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *Comput. Surveys* 55, 6 (2022), 1–28.

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1101–1110.

[33] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[34] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1120–1128.

[35] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.

[36] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui. 2020. Graph neural networks in recommender systems: a survey. *arXiv preprint arXiv:2011.02260* (2020).

[37] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[38] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Mengyin Lu, and Liefeng Bo. 2021. Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1931–1936.

[39] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4486–4493.

[40] Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. 2022. Multi-Behavior Sequential Recommendation with Temporal Graph Transformer. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[41] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems.. In *IJCAI*, Vol. 17. Melbourne, Australia, 3203–3209.

[42] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2263–2274.

[43] Jaehyuk Yi and Jinkyoo Park. 2020. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3366–3376.

[44] Ying Yin, Li-Xin Ji, Jian-Peng Zhang, and Yu-Long Pei. 2019. DHNE: Network representation learning method for dynamic heterogeneous networks. *IEEE Access* 7 (2019), 134782–134792.

[45] Enming Yuan, Wei Guo, Zhicheng He, Huifeng Guo, Chengkai Liu, and Ruiming Tang. 2022. Multi-Behavior Sequential Transformer Recommender. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1642–1652.

[46] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.

[47] Yuxuan Zhou, Chao Li, Zhi-Qi Cheng, Yifeng Geng, Xuansong Xie, and Margret Keuper. 2022. Hypergraph Transformer for Skeleton-based Action Recognition. *arXiv preprint arXiv:2211.09590* (2022).