Towards Efficient Hypergraph Representation Learning for Multi-Behaviour Recommender Systems

Tendai Mukande
Research Ireland ML-LABS
Dublin City University
Dublin, Ireland
tendai.mukande@mail.dcu.ie

Esraa Ali
ADAPT Centre
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
Insight Research Ireland Centre for Data Analytics
University College Dublin
Dublin, Ireland
ruihai.dong@ucd.ie

Noel O'Connor
Insight Research Ireland Centre for Data Analytics
Dublin City University
Dublin, Ireland
Noel.OConnor@dcu.ie

ABSTRACT

Recommender systems (RS) are essential for the modern web, providing personalized suggestions that alleviate information overload and enhance the user experience across various platforms. Graph neural networks (GNN) have been proposed for RS and demonstrate significant potential. However, GNN-based methods are prone to oversmoothing in practical settings, limiting their expressive power and ability to capture complex data patterns effectively [15]. Recent research has also explored graph-transformer-based RS methods that, while improving performance, tend to increase computational costs, particularly in large-scale scenarios. To address these challenges, we introduce FAHMRec, an efficient hypergraph-based model for multi-behaviour recommendation. Experimental results demonstrate that our method outperforms state-of-the-art baselines in recommendation quality while also reducing memory and time costs.

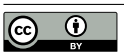
CCS CONCEPTS

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

KEYWORDS

Transformer; Multi-Behaviour Recommender System; MLP Hypergraph; Graph Neural Network; Computational Efficiency.

ACM Reference Format:
Tendai Mukande, Esraa Ali, Annalina Caputo, Ruihai Dong, and Noel O'Connor. 2025. Towards Efficient Hypergraph Representation Learning for Multi-Behaviour Recommender Systems. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715572



WWW Companion '25, April 28-May 2, 2025, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/25/04.
https://doi.org/10.1145/3701716.3715572

1 INTRODUCTION

Recommender systems have emerged as an effective solution to mitigate information overload on various web platforms. In real-world scenarios where data continually evolves, developing efficient models to generate high-quality recommendations remains a critical research challenge. Graph neural networks have been proposed to enhance recommendation systems by leveraging the message-passing scheme [4] to model user-item interactions. However, message-passing GNN methods face several challenges, including oversmoothing and oversquashing [2]. *Oversmoothing* occurs when node embeddings converge and become nearly indistinguishable after multiple layers of message passing, which leads to loss of node information, hindering the model's ability to capture long-term dependencies within the graph [5]. On the other hand, *oversquashing* arises when information transfer between distant nodes is insufficient due to repeated message-passing steps [17, 28]. Consequently, signals from distant nodes are progressively diluted, reducing the model's ability to represent complex behaviours and long-range interactions. Furthermore, GNN methods struggle to effectively model dependencies that extend beyond the reach of their message-passing framework [15]. Moreover, the self-attention mechanism [24] applied in most GNN-based RS methods, adds significant computational overheads due to its quadratic time and space complexity with respect to input size, which makes these methods less practical for large-scale applications [23]. These challenges pose an important question: *How can we improve the computational efficiency of GNN-based RS methods without loss of recommendation accuracy in large-scale scenarios?*

Motivated by this question, we propose the **Fast Attention Hypergraph Multi-Behaviour Recommendation (FAHMRec)** model, which aims to simultaneously address both performance and efficiency issues. Our **contributions** are summarized as follows: We propose a novel hypergraph representation learning model for multibehaviour recommendation, which leverages high-order permutation-invariant layers without message-passing. The model effectively captures user-item interactions across diverse user behaviours, ensuring consistent learning while maintaining robust output. This representation learning approach has been shown to

be more expressive compared to the GNN message passing scheme [9, 22]. Additionally, we introduce an efficient higher-order attention mechanism that approximates standard self-attention with linear time and space complexity, thereby significantly reducing memory usage and achieving faster model runtime. Experimental results using three real-world datasets demonstrate improvement in both performance and computational efficiency compared to state-of-the-art methods.

2 RELATED WORK

Hypergraph neural networks (HGNN) have emerged as a powerful tool in representation learning to address the limitations of traditional GNN methods [1]. HGNNs have been applied to model higher-order relationships to capture complex interactions more effectively through nodes and hyperedges [11]. The use of HGNNs to improve the performance of recommendation is an active research area [12, 30]. Taking into account multiple items and users simultaneously, HGNNs have been shown to capture more nuanced interactions, leading to better quality recommendations [14].

The success of transformers has driven their adoption in various tasks, including recommender systems [12]. However, their high computational cost remains a significant bottleneck for scalability [13]. To address this, several techniques have been proposed to enhance transformer efficiency. These include low-rank approximations [18, 23], kernel-based approaches [18] and sparse attention methods that reduce computation time and memory usage [13]. In this work, we also propose an efficient attention framework to address scalability challenges in recommender systems [19].

Recent research has explored the representation of GNNs as MLP-based linear layers [10], offering a new research direction for more efficient representation learning models that process graph data without relying on the message-passing scheme [22]. Maron et al. introduced linear operators between tensors of arbitrary order, demonstrating that their method is more expressive than several message-passing GNN baselines [16]. Similarly, Kim et al. also showed that GNNs could be represented as independent tokens and embedded as input to transformers, proving that this approach outperforms message passing in terms of expressiveness [8]. Inspired by this approach, we extend the non-message passing paradigm to develop faster and more scalable recommendation systems.

3 METHODOLOGY

In this section, we introduce the FAHMRec model. We apply high-order linear layers to represent user-item interactions, and kernel attention mechanisms to capture complex dependencies, reducing the computational complexity from quadratic to linear. The model architecture is shown in Figure 1.

3.1 Hypergraph Representation

We define a higher-order tensor $A \in \mathbb{R}^{n^k \times d}$, where k is the order of the tensor, d is the input feature dimension and d' is the output feature dimension.

Embedding Layer: We define a higher-order linear layer $L_{k\to l} : \mathbb{R}^{n^k \times d} \to \mathbb{R}^{n^l \times d'}$, which projects A into a new feature space:

$$L_{k\to l}(A)_j = \sum_i B_{i,j} A_i W, \tag{1}$$

where: $A \in \mathbb{R}^{n^k \times d}$ is the input tensor, representing the features of hyperedges, $B_{i,j} \in \mathbb{R}^{n^{k+l}}$ is the weight matrix that encodes interactions between nodes, $W \in \mathbb{R}^{d \times d'}$ is a learnable weight matrix.

3.2 Attention Module

To improve computational efficiency, we compute the attention of the transformed tensor $L_{k\to l}(A)$, reducing the complexity from quadratic to linear. The output of self-attention is computed using the following equation:

$$\mathrm{Attn}_{k\to l}(A)_j = \sum_{h=1}^{H} \sum_i \alpha_{i,j}^h \big(L_{k\to l}(A)_i\big) W_h^V W_h^O, \tag{2}$$

where: H is the number of attention heads, $L_{k\to l}(A)_i \in \mathbb{R}^{d'}$ is the i-th row of the transformed tensor, $W_h^V \in \mathbb{R}^{d' \times d_v}$ and $W_h^O \in \mathbb{R}^{d_v \times d_o}$ are the value and output projection matrices and $\alpha_{i,j}^h$ is the attention coefficient for the h-th head, computed as:

$$\alpha_{i,j}^h = \frac{\exp(\sigma(Q_j^h, K_i^h)) \cdot S_{i,j}}{\sum_{i' \in E} \exp(\sigma(Q_j^h, K_{i'}^h) \cdot S_{i',j})}. \tag{3}$$

where: Q_j^h and K_i^h are the query and key vectors for the j-th output and i-th input, respectively, and $\sigma(Q_j^h, K_i^h)$ measures their similarity. $S_{i,j} \in \{0, 1\}$ represents active hyperedges, with $S_{i,j} = 1$ indicating a valid connection between i and j. Restricting computations to non-zero entries of S reduces complexity from $O(n^2)$ to $O(k \cdot n)$, where k is the number of active hyperedges per query. Thus, where n is the number of nodes, the overall computational complexity is reduced to linear:

$$O(n \cdot d \cdot d') + O(k \cdot n \cdot d') \approx O(k \cdot n \cdot d') \quad \text{where} \quad k \ll n, \tag{4}$$

3.3 Model Training

Our training objective is to minimize the cross-entropy loss. Let $P(j|u, i)$ denote the predicted probability that item j is the next item for user u after interacting with item i. Given a set of items $J = \{j_1, j_2, \ldots, j_C\}$, we define the cross-entropy loss as follows:

$$L_{\mathrm{CE}} = -\frac{1}{N} \sum_{(u,i) \in \mathcal{D}} \sum_{j \in J} y_{u,j} \log P(j|u, i) \tag{5}$$

where N is the total number of user-item pairs in the data $\mathcal{D}$, $y_{u,j}$ is a binary indicator that is 1 if the next item for user u is j and 0 otherwise.

4 EXPERIMENTS

In conducting our experiments, our objective is to answer the following research questions: **RQ1:** Does the proposed model improve the performance of the recommendation? **RQ2:** Does the proposed framework reduce the memory and runtime cost of the model?

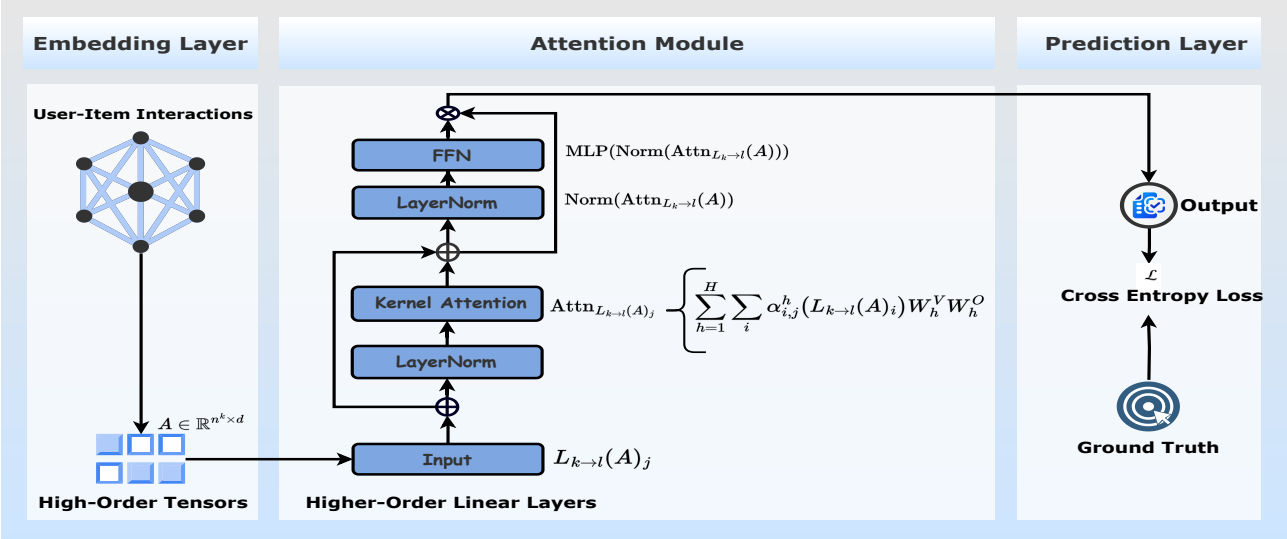


Figure 1: Proposed FAHMRec framework. User-item interactions are embedded as higher-order tensor representations. Kernel attention is then applied to capture complex user-item interactions.

Dataset	Metric	Model									
		SASRec	BERT4Rec	MB-GCN	MB-GMN	KMCLR	MB-STR	MBHT	HMAR	PBAT	FAHMRec
Taobao	HR@5	0.257	0.275	0.244	0.398	0.459	0.694	0.682	0.692	0.737	**0.756***
	NDCG@5	0.186	0.197	0.178	0.226	0.277	0.583	0.594	0.593	0.650	**0.677***
	HR@10	0.379	0.395	0.367	0.496	0.575	0.776	0.768	0.819	0.805	**0.826***
	NDCG@10	0.227	0.245	0.214	0.306	0.387	0.616	0.608	0.658	0.676	**0.705***
IJCAI	HR@5	0.486	0.496	0.348	0.463	-	0.802	0.776	-	0.874	**0.885***
	NDCG@5	0.356	0.366	0.212	0.289	-	0.695	0.678	-	0.784	**0.816***
	HR@10	0.602	0.625	0.451	0.522	-	0.884	0.854	-	0.914	**0.932***
	NDCG@10	0.415	0.428	0.271	0.339	-	0.716	0.707	-	0.795	**0.809***
Tianchi	HR@5	0.584	0.594	0.625	0.655	0.689	0.746	0.769	0.775	0.789	**0.814***
	NDCG@5	0.349	0.353	0.426	0.454	0.483	0.496	0.488	0.495	0.567	**0.603***
	HR@10	0.606	0.684	0.728	0.735	0.712	0.756	0.782	0.795	0.813	**0.847***
	NDCG@10	0.477	0.492	0.505	0.539	0.535	0.593	0.582	0.623	0.618	**0.639***
Efficiency Evaluation: Runtime (sec/epoch) and Memory (GB) for NDCG@10 and HR@10											
Taobao	Memory (T)	12.72	12.84	13.24	14.33	14.56	15.44	11.12	12.24	13.45	**10.26**
	Runtime (T)	1798.45	1886.14	2135.22	2306.94	2387.27	2616.65	1608.53	1672.79	1976.82	**1555.12**
	Memory (I)	12.56	12.84	13.39	14.65	14.73	15.67	11.71	12.65	13.72	**10.34**
	Runtime (I)	8.54	17.96	19.25	18.54	11.75	10.67	8.53	7.52	8.67	**7.46**
IJCAI	Memory (T)	13.87	13.94	11.82	14.98	-	15.87	11.78	-	14.43	**10.86**
	Runtime (T)	1923.15	1972.21	2242.37	2368.46	-	2689.13	1669.62	-	2051.27	**1061.90**
	Memory (I)	13.94	14.02	12.22	15.16	-	16.07	11.86	-	14.77	**11.34**
	Runtime (I)	12.31	21.59	25.61	24.39	-	12.93	11.72	-	10.18	**9.27**
Tianchi	Memory (T)	8.34	8.89	9.23	10.01	11.16	12.80	8.19	10.27	9.64	**8.09**
	Runtime (T)	1522.69	1675.08	1888.74	1992.30	2006.74	2125.49	1487.62	1573.54	1678.82	**1412.82**
	Memory (I)	8.54	8.91	9.34	10.60	11.38	12.96	8.24	10.29	9.78	**8.42**
	Runtime (I)	6.36	15.52	18.92	17.46	9.46	8.53	6.42	5.12	8.67	**4.46**

Table 1: Performance comparison: The best performances indicated in bold show the relative improvement over the best performing baseline at 0.05 significance with paired t-test. The underlined values indicate the second best performance. For efficiency evaluation, the memory usage and runtime are measured during training (T) and inference (I).

Experimental Setup. For performance evaluation and fair comparison, we use the leave-one-out strategy and follow the Cloze task settings in [27] and [21]. We obtain the test samples from the last purchase for each user and the validation samples from the previous purchases. Grid search is used to select hyperparameter values. The input feature dimension (d) and the output feature dimension (d') are tested within the range $\{64, 128, 256\}$. The number of attention heads (H) is varied between $\{4, 8, 16\}$. The depth of the embedding module is tested for $\{1, 2, 3\}$ layers, and the learning rate is tuned from $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ with a cosine decay scheduler. The batch size is varied on $\{128, 256, 512\}$, and the dropout rate is tested within $\{0.0, 0.1, 0.2\}$. The number of active hyperedges per query (k) is explored within $\{5, 10, 20\}$. Regularization is controlled using a weight decay factor (λ) chosen from $\{1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}\}$. Finally, the AdamW optimizer is used for parameter updates. Our model is implemented using the Pytorch RecBole framework [31].

Dataset	#Users	#Items	#Interactions	Behaviour Type
Taobao	147,894	99,037	7,658,926	[view, fav, cart, buy]
IJCAI	423,423	874,328	36,222,123	[view, fav, cart, buy]
Tianchi	25,000	500,900	4,619,389	[view, fav, cart, buy]

Table 2: Summary of dataset statistics.

Datasets. For our experiments, we use three datasets summarized in Table 2. The **Taobao** dataset contains *purchase, add-to-favourites, add-to-cart* and *view* user-item interactions [20]. **IJCAI** was released for the IJCAI 2015 contest and contains *purchase, add-to-favourites, add-to-cart,* and *view* user-item interactions [20]. **Tianchi**, collected from Tmall, includes *buy, add-to-cart, add-to-favourite* and *view* behaviours [3].

Baselines. We evaluate our model by comparing with the following baselines:

- **SASRec**: A sequential recommendation model that uses the transformer self-attention mechanism to identify relevant items at each time step for the next-item prediction task [7].
- **BERT4Rec-M**: A sequential model that applies bi-directional transformer-based self-attention for recommendation [21].
- **MB-GCN:** A multi-behaviour graph convolutional network-based model which combines behaviour-aware user-to-item embedding propagation as well as item-to-item embedding propagation layers for multi-behaviour recommendation [6].
- **MB-GMN:** Integrates graph neural networks with meta-networks for multi-behaviour recommendation [25].
- **KMCLR:** Applies a multi-behaviour learning module to extract user information and uses a knowledge graph to derive item representations for recommendation [26].
- **MBHT:** A multi-behaviour recommendation model that applies a multi-scale transformer to capture local user-item dependencies and a hypergraph convolution module to capture global dependencies [27].
- **MB-STR:** A multi-behaviour model which integrates a behaviour transformer layer and a sequential pattern generator module for sequential recommendation [29].
- **HMAR:** Uses masked self-attention to items of the same behaviour, and across behaviours, and encodes the historical frequency of each item's behaviour in the input sequence for multi-behaviour recommendation [3].
- **PBAT:** A multi-behaviour recommendation model which applies a personalised behaviour pattern generator to extract dynamic behaviour patterns [20].

Dataset	Metric	Model Variant			
		FAHMRec	**FAHM-V**	**FAH-MP**	**FAHM-SB**
Taobao	HR@10	**0.826**	0.817	0.724	0.621
	NDCG@10	**0.705**	0.697	0.610	0.532
	Memory (T)	10.26	13.45	12.46	**9.76**
	Runtime (T)	1555.12	2352.11	2140.89	**1462.09**
	Memory (I)	10.34	13.35	11.67	**8.75**
	Runtime (I)	7.46	10.21	9.98	**7.20**
IJCAI	HR@10	**0.932**	0.928	0.891	0.816
	NDCG@10	**0.809**	0.801	0.731	0.683
	Memory (T)	10.86	13.95	12.23	**10.10**
	Runtime (T)	1061.94	1876	1398.69	**1005.09**
	Memory (I)	11.34	13.35	11.43	**10.58**
	Runtime (I)	9.27	11.76	10.98	**8.82**
Tianchi	HR@10	**0.847**	0.826	0.746	0.723
	NDCG@10	**0.639**	0.631	0.582	0.554
	Memory (T)	8.09	12.45	11.68	**7.96**
	Runtime (T)	1412.82	2531.05	1740.54	**1360.32**
	Memory (I)	8.42	10.35	9.57	**8.21**
	Runtime (I)	4.46	6.76	5.98	**4.22**

Table 3: Ablation study results.

Metrics. We evaluate the performance of our model using the following metrics: HR@k and Normalised Discounted Cumulative Gain (NDCG@k) where k is set to 5 and 10. We also evaluate the memory usage and run-time.

4.1 Performance Evaluation (RQ1)

Experimental results in Table 1 show that incorporating multi-behaviour information improves the recommendation performance, as validated by the superior performance of multi-behaviour models MBHT, HMAR and PBAT compared to the sequential methods SASRec and BERT4Rec. Our model outperforms both sequential and message-passing hypergraph methods, which highlights the efficacy of FAHMRec in capturing complex higher-order interactions, thus improving recommendation performance.

Ablation Study: We study the influence of our model components by introducing variants that do not incorporate the respective modules. **FAHM-SB:** we remove the collaborative multi-behaviour information and use only purchases for model evaluation, **FAHM-MP:** we apply a hypergraph convolution network in place of the MLP hypergraph module , and **FAHM-V:** we use the standard vanilla self-attention instead of kernel attention. As shown in Table 3, all three variants of the FAHMRec model show a degradation in performance against the complete FAHMRec model, which further validates the benefits of the MLP hypergraph, efficient attention and multi-behaviour modelling. The degraded performance against the FAHMRec model validates that our proposed method is more expressive than message passing. Another observation is that FAHMRec model variants show better performance compared

to the sequential methods SASRec and BERT4Rec. We attribute this to the contribution of the hypergraph module, which enables higher-order representation learning.

4.2 Efficiency Evaluation (RQ2)

We study the computational efficiency of our model by comparing memory and runtime with the baseline models. From the results in Tables 1 and 3, our model shows superior efficiency compared to the vanilla attention-based models SASRec, BERT4Rec, MB-STR, PBAT and the FAHM-V variant, and the GNN message passing methods MB-GCN and MB-GMN. Although performance is similar between the FAHMRec and FAHM-V model variants, the former is significantly more efficient in terms of memory usage and runtime. This validates the efficiency of our kernel-based attention module compared to standard self-attention. MBHT shows relatively better efficiency compared to other baselines, which can be attributed to the low-rank attention used in the model.

5 CONCLUSION

In this paper, we introduce the FAHMRec model that uses a non-message passing hypergraph model to capture intricate user-item interactions for multi-behaviour recommendation. Experimental evaluations on three real-world e-commerce datasets demonstrate the superior performance of our proposed model compared to state-of-the-art baselines while requiring less memory and achieving faster runtime. These results highlight the efficacy of our proposed approach in addressing the challenges posed by heterogeneous data, paving the way for more effective and scalable learning in real-world recommender systems.

ACKNOWLEDGMENTS

This research is supported by Taighde Éireann – Research Ireland under Grant number 18/CRT/6183.

REFERENCES

[1] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. 2023. A survey on hypergraph representation learning. *Comput. Surveys* 56, 1 (2023), 1–38.

[2] Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. 2022. Preventing over-smoothing for hypergraph neural networks. *arXiv preprint arXiv:2203.17159* (2022).

[3] Shereen Elsayed, Ahmed Rashed, and Lars Schmidt-Thieme. 2024. HMAR: Hierarchical Masked Attention for Multi-behaviour Recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 131–143.

[4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[5] Jhony H Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D Malliaros. 2023. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 566–576.

[6] 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.

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

[8] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. 2022. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems* 35 (2022), 14582–14595.

[9] 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.

[10] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. 2018. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144* (2018).

[11] Yinfeng Li, Chen Gao, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1997–2002.

[12] Bingqian Liu, Duantengchuan Li, Jian Wang, Zhihao Wang, Bing Li, and Cheng Zeng. 2024. Integrating user short-term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation. *Information Processing & Management* 61, 3 (2024), 103680.

[13] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. 2023. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14420–14430.

[14] Ang Ma, Yanhua Yu, Chuan Shi, Zirui Guo, and Tat-Seng Chua. 2024. Cross-view hypergraph contrastive learning for attribute-aware recommendation. *Information Processing & Management* 61, 4 (2024), 103701.

[15] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. 2019. Provably powerful graph networks. *Advances in neural information processing systems* 32 (2019).

[16] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. 2018. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902* (2018).

[17] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* 35 (2022), 14501–14515.

[18] 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.

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

[20] Jiajie Su, Chaochao Chen, Zibin Lin, Xi Li, Weiming Liu, and Xiaolin Zheng. 2023. Personalized behavior-aware transformer for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6321–6331.

[21] 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.

[22] Bohan Tang, Siheng Chen, and Xiaowen Dong. 2024. Hypergraph-MLP: learning on hypergraphs without message passing. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 13476–13480.

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

[24] 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).

[25] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 757–766.

[26] Hongrui Xuan, Yi Liu, Bohan Li, and Hongzhi Yin. 2023. Knowledge Enhancement for Contrastive Multi-Behavior Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 195–203.

[27] 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.

[28] Zhouxin Yu, Jintang Li, Liang Chen, and Zibin Zheng. 2022. Unifying multi-associations through hypergraph for bundle recommendation. *Knowledge-Based Systems* 255 (2022), 109755.

[29] 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.

[30] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. 2021. Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 2557–2567.

[31] 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.