2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

PolyRec : Polynomial Attention for Enhanced Sequential Recommendation

Peichen Ji[1,2], Jiwei Qin✉[1,2], Jie Ma[1,2] and Yanping Chen[3]

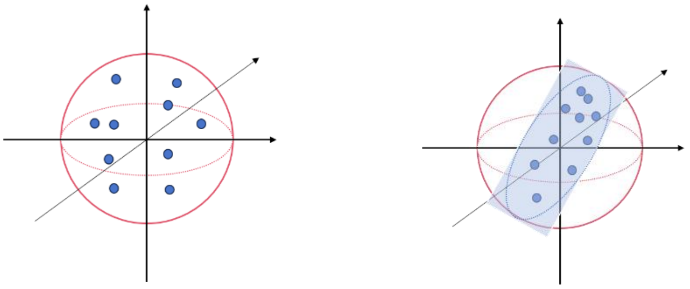
[1] School of Computer Science and Technology, Xinjiang University, Urumqi, China
[2] Key Laboratory of Signal Detection and Processing, Xinjiang University, Urumqi, China
[3] College of Computer Science and Technology, Guizhou University, Guiyang, 550025, Guizhou, China
✉: `jwqin@xju.edu.cn`

Abstract. The self-attention mechanism has been widely adopted in sequential recommendation due to its powerful capability in modeling long-range dependencies. However, as the number of attention layers increases, user embedding vectors tend to collapse into a low-dimensional subspace. This collapse of embedding space leads to overly concentrated user distributions, resulting in the dimensional collapse phenomenon. The increased similarity among user embedding representations makes it challenging for the model to distinguish between different users, ultimately causing recommendation results to become homogenized. To mitigate this issue, we propose a novel sequential recommendation model named Polynomial Attention for enhanced Sequential Recommendation (PolyRec), which alleviates spatial collapse and improves the distribution of user representations. Firstly, the model can better capture high-order structural information through the incorporation of high-order polynomial terms. Simultaneously, leveraging the orthogonality and optimal approximation properties of Chebyshev coefficients stabilizes the parameter training process and enhances the representation capability of the attention mechanism. Furthermore, we conduct a theoretical analysis to demonstrate that during neural networks' aggregation of target information, feature representations are prone to being squeezed by noise and redundant information, thereby exacerbating dimensional collapse. Therefore, by introducing Fourier transforms, we truncate the traditional residual connections in the frequency domain. This approach effectively retains more important information, thereby alleviating the over-squashing phenomena. Experimental evaluations on four datasets demonstrate that PolyRec outperforms other baseline methods in recommendation accuracy.

Keywords: Sequential recommendation, Dimensional collapse, Polynomial.

1 Introduction

Recently, sequential recommendation have been widespread use across various online platforms to predict users' future interests in products and videos[10]. In real-world recommendation systems, user behavior is fluid and evolves over time. The primary advantage of sequential recommendation model is their ability to explicitly model item sequences over time, enabling more accurate capture of dynamic information in user sequences. The Transformer model[20] assign attention weights to capture the correlations between items. Therefore, Transformer-based algorithms can accurately capture the dynamic information within user sequences and have been widely applied in sequential recommendation. Although Transformer models have achieved remarkable performance, as the number of layers increases, the embedding vectors in the sequence model rapidly converge to a uniform or balanced state. In actual model training, this phenomenon is referred to as dimension collapse. This convergence towards uniform token representations diminishes the model's expressive capacity because the resulting low-rank matrices are unable to capture the complex relationships between tokens. Additionally, this dimension collapse may lead to gradient vanishing and potential instability during the training process[6].



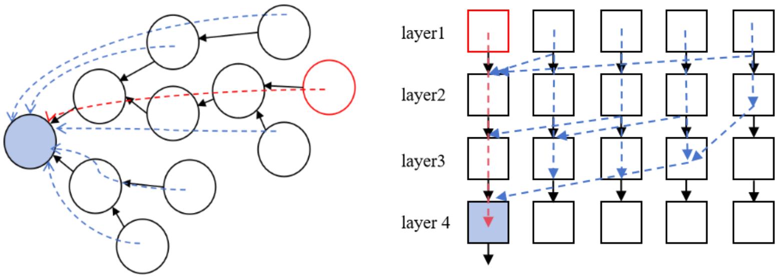
(a) Ideal representation

(b) Dimensional collapse

Fig. 1. (a) The embedding distribution spans the entire high-dimensional space. (b) The embedding distribution collapses into a narrow plane.

In neural network-based models, the dimensional collapse problem may originate from excessive compression of information during transmission. In Recurrent Neural Network(RNN) based sequential recommendation algorithms, as the number of time steps increases, users' recent behaviors excessively compress the information from earlier actions. This makes it difficult for the model to effectively remember and utilize the key information from the earlier parts of the sequence, a phenomenon known as long-term forgetting. In GNN-based algorithms, the number of GNN layers is stacked to mine information from distant nodes. However, after continuous message passing and aggregation, node features become over-compressed, resulting in the loss of key information and a decline in expressive capacity. As the number of Transformer layers increases, while extracting deeper features, noise and irrelevant information are also introduced. This may cause the original information to be gradually diluted or lost, preventing the sufficient preservation of the unique characteristics of nodes. Vaswanic et al.[21] conducted a systematic study of the Transformer architecture in response to dimensional collapse, both theoretically and experimentally, showing that the residual connections and layer normalization components in the architecture help mitigate this issue. Residual connections have been proven in the past to improve the propagation of deep signals. In our approach, we further optimize training and reduce parameter compression by introducing Fourier transforms into the residual connections, preserving more important signals in the frequency domain and employing truncated residual connections.

Although Transformer models hold great potential, their expressive power is limited by the collapse of the self-attention matrix, which prevents the model from fully realizing its performance. When the representations of the entire sequence collapses into a lower-dimensional space, the learning capacity of the encoder is severely wasted. We observe that the low-rank and collapse phenomena in the attention matrix lead to rank collapse and gradient vanishing, further hindering the model's performance improvement.



(a) GNN (b) Transformer

Fig. 2. Over-squashing phenomena in common neural networks. When aggregating target information (red nodes), the neural network is squeezed by noise and redundant information (blue nodes), resulting in an information bottleneck.

To enhance the representational capacity of the self-attention matrix, we draw inspiration from Graph Signal Processing (GSP). In the spectral domain theory of graph neural networks, graph filters are typically defined by the adjacency matrix or Laplacian matrix. Based on related research, we treat the traditional self-attention mechanism as a normalized adjacency matrix and extend it using graph filter theory, which introduces greater flexibility and scalability to self-attention. Based on this, we propose a polynomial attention mechanism based on matrix polynomials to replace the traditional single-matrix-based attention approach, thereby improving the model's capabilities in both encoding and representation layers.

Our contributions are as follows:

1. We reveal that Transformer-based algorithms in recommendation systems suffer from the issue of dimensional collapse, which leads to insufficient user representation and prevents the model from fully exploiting its potential performance.

2. To enhance the representational capacity of the attention mechanism, we propose a sequential recommendation model based on polynomial matrices. We draw inspiration from the Chebyshev polynomials in Graph Signal Processing to strengthen the attention mechanism's representational effectiveness. Additionally, we introduce Fourier transform into the residual connections, using truncated residual connections to more effectively stabilize gradient training.

3. We conduct experiments on four widely-used datasets in recommendation systems. The results show that our model significantly improves the representational capacity of the attention mechanism and effectively alleviates the dimensional collapse problem.

2 Related Work

2.1 Sequential Recommendation

Early methods used Markov chain[8] to model user behavior, followed by deep learning approaches like GRU4Re[9] and Caser[19] for feature extraction. With the success of attention mechanisms, models like SASRec[12], BERT4Rec[18], and S3Rec[25] applied attention mechanism in sequential recommendations. Despite their potential, attention-based models still face challenges such as data sparsity and overfitting. To address this, contrastive learning has been integrated, with models like CL4SRec[24], CoSeRec[13], and DuoRec[15] using data augmentation and contrastive strategies to improve performance. BSARec[16] and ADAmct[11] address overfitting through inductive biases.

2.2 Dimensional collapse

Unlike convolutional neural networks ,Transformer models exhibit performance saturation as the number of layers increases, leading to a phenomenon called dimensional collapse. This occurs when the embedding vectors collapse into a low-dimensional subspace, rather than utilizing the entire available embedding space. The singular values of the embedding matrix rapidly decrease to minimal values, leaving all but the dominant dimensions ineffective, resulting in a matrix with very low rank. Researchers have analyzed the issue of self-attention maps converging into low-rank matrices, especially with attention scores concentrating on meaningless tokens such as [CLS] and [SEP]. Similar issues have been observed for visual tasks, with several studies addressing the collapse of self-attention in Transformer encoders.

Dong et al. [6] proposed that the cause of dimensional collapse lies in the row-random nature of the attention matrix, and used path decomposition to shown that adding skip connections and multilayer perceptrons helps prevent this collapse. Noci et al.[14] further demonstrated that rank collapse is not only a problem during inference, but also affects training due to gradient vanishing issues arising from initialization. Wu et al.[23] further studied the impact of LayerNorm on rank collapse in Transformers and proved that these components help prevent or mitigate rank collapse.

2.3 Graph transformer

GNNs can handle non-Euclidean data through message passing and neighbor aggregation. Injecting topological properties of the graph for Tranformer-based models allows the model to have a prior on structural position in high-dimensional space. Therefore, researchers have introduced GNNs into sequential recommendation. Early Graph

Transformers were mainly used by simply cascading GNNs and Transformers[2]. In subsequent research, more fine-grained coding information is introduced to fuse local and global information more systematically and flexibly[22][4][5], such as positional encoding(PE), relational encoding(RE), structural encoding(SE).

3 Method

We begin to introduce the details of PolyRec. First, we use an embedding layer to map the items into vectors. Then, we use polynomial attention as the backbone network. Next, we enhance the model's representation with truncated residual connections. Finally, we also employ contrastive learning to provide extra supervision information for the model.

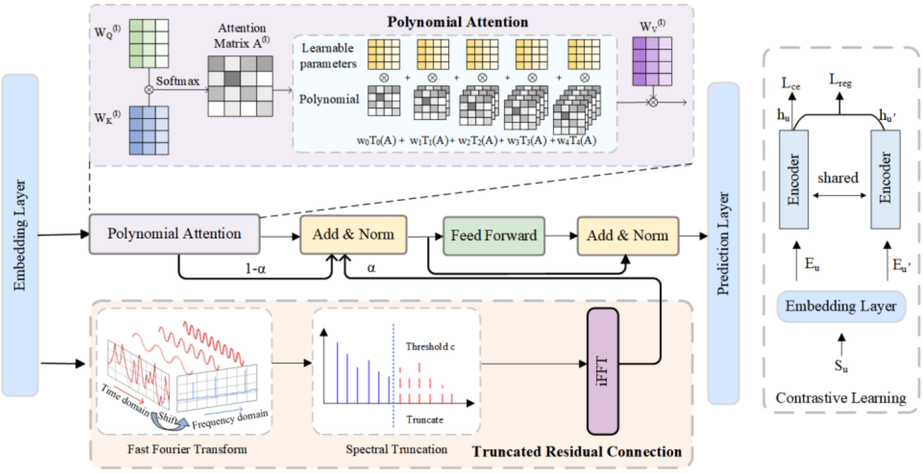


Fig. 3. An overview of PolyRec.

3.1 Embedding Layer

User's historical interaction sequences are embed as multidimensional vectors for subsequent modelling. In PolyRec, $\mathrm{i} \in I$ symbolizes item and $\mathrm{u} \in \mathcal{U}$ symbolizes user. Item IDs are labeled as $I = \{i_1, i_2, \dots, i_{|I|}\}$ and the user IDs are labeled as $\mathcal{U} = \{\mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_{|u|}\}$. Additionally, we add t as timestep in order to reflect the temporal nature of user behavior. The user behavior sequence is represented as $\mathrm{S_u} = [i_1^{(u)}, i_2^{(u)}, \dots, i_t^{(u)}, \dots, i_N^{(u)}]$, where $\mathrm{S_u} \in S$, $\mathrm{u} \in \mathcal{U}$, $i_t^{(u)} \in I$, and N is the length. For S_u can be denoted as:

$$E_u = [e_1^{(u)}, e_2^{(u)}, \dots, e_N^{(u)}] \tag{1}$$

We use position encoding P to add extra positional information to the model. In addition, we also apply dropout and layer normalization operations as follows:

$$E_u = \text{Dropout}\big(\text{ LayerNorm } (E_u + P)\big) \tag{2}$$

Although each item has an ID that is different from each other, similar items may have the same eigenvalues after the embedding layer.

3.2 Polynomial Attention Encoder

The primitive attention mechanism encodes the relationships between sequence elements by forming an attention weight matrix A via Q and K vectors, thus highlighting important information in the sequence. However, a single attention weight matrix A has limited representational power, and we consider the traditional self-attention mechanism as the most basic graph filter to improve the representational power of the primitive attention matrix A by introducing spectral graph theory and using a self-attention layer based on Chebyshev polynomials. The polynomial self-attention layer consists of learnable parameters and polynomial coefficients T_k containing higher order terms A^k, where k is the order of the polynomial. The polynomials can better capture higher-order structural information through the higher-order terms. Moreover, the orthogonality and best approximation properties of the Chebyshev polynomial coefficients help stabilize parameter training, effectively mitigating the dimensional collapse phenomenon.

Attention mechanisms. The attention score matrix A is obtained by calculating the dot product of the Q and the K vector.

$$\text{Output } = \text{ Attention-Score } (Q, K) \cdot V = A \cdot V \tag{3}$$

In PolyRec, we replace the attention matrix A with a polynomial attention layer H that includes A.

Graph signal filtering. GNNs can handle non-Euclidean data. We can understand GNNs from two perspectives: spectral domain GNNs and spatial domain GNNs. Spectral domain GNNs, based on graph signal filtering and spectral graph theory, define graph convolution from the perspective of the spectral domain. Spectral domain graph convolution introduces graph Fourier transform to convert graph signals from the spatial domain to the frequency domain. The eigenvectors of the Laplacian matrix are used as the basis for graph Fourier projections. For undirected graphs, it can be decomposed as follows:

$$L = U \Lambda U^T \tag{4}$$

Each column of $U \in \mathbb{R}^{N \times N}$ is an eigenvector of L , and $\Lambda \in \mathbb{R}^{N \times N}$ is a diagonal matrix. Given a graph signal $x \in R^N$ on graph G, its graph Fourier transform is:

$$\widehat{x_k} = \sum_{i=1}^{N} U_{ki}^T x_i \tag{5}$$

It is represented in matrix form as:

$$\hat{x} = U^T x \tag{6}$$

Correspondingly, the graph inverse Fourier transform is represented as:

$$x = U\hat{x} \tag{7}$$

Thus, the convolution operation in the spectral domain can be represented as:

$$\begin{aligned} x * g &= U\big((U^T x) \odot (U^T g)\big) \\ &= U(U^T g \odot U^T x) \end{aligned} \tag{8}$$

Where g is the convolution kernel, and the Hadamard product is used as $\odot$. If we treat the entire operation as a learnable convolution kernel $U^T g$, denoted as $g_\theta(\Lambda)$, the final convolution formula is:

$$y = U g_\theta(\Lambda) U^T x \tag{9}$$

Polynomial enhanced attention. We define A as a self-attention matrix, where n is the number of tokens, and d is the embedding dimension. V is the value matrix. We treat the self-attention matrix as a special normalized adjacency matrix and view the self-attention computation process as a simplified graph filtering process. Therefore, we extend the self-attention using matrix polynomials, and the extended version can be represented as:

$$HV = \sum_{k=0}^{n-1} w_k \mathrm{A}^k \mathrm{V} \tag{10}$$

Where w are the polynomial coefficients, the self-attention based on matrix polynomials can be expressed as:

$$HV \approx w_0 \mathrm{V} + w_1 AV + w_2 \mathrm{A}^2 \mathrm{V} + \cdots + w_j \mathrm{A}^j \mathrm{V} \tag{11}$$

Where j is the tolerable error convergence range. In addition, spectral domain GCN constructs graph filters using Chebyshev polynomials for approximation. Since Chebyshev polynomials have orthogonality and stability, they ensure better numerical stability during computations. We generate the Chebyshev coefficients recursively $T_{n+1}(A) = 2AT_n(A) - T_{n-1}(A)$, so that we do not introduce excessive computational effort during training. The final output of the polynomial attention layer is expressed as follows:

$$HV \approx w_0 T_0(\mathrm{A})\mathrm{V} + w_1 T_1(\mathrm{A})\mathrm{V} + \cdots + w_j T_j(\mathrm{A})\mathrm{V} \tag{12}$$

Truncated residual connection. In PolyRec, we use truncated residual connections to optimize the information flow. Unlike standard residual connections, we shift the perspective to the frequency domain by introducing a Fourier transform, which performs spectral truncation in the frequency domain. With spectral truncation, we force the model to focus on learning the most important low-frequency information while eliminating the high-frequency noise, thus retaining the more important elements of the sequence more efficiently. In previous work, feature reuse is usually achieved by adding additional connections, such as DenseNet, which directly connects each layer to all previous layers. Alternatively, more complex neural networks are added with the introduction of gating or convolutional kernels to extract features. However, the above methods usually introduce additional computational complexity. Truncating the residual connections can reduce the computational burden by effectively compressing the information through frequency domain computation. Mathematically, the residual signals are processed by Fourier transform, and the global convolution is realized by frequency domain product. By learning the correlation of different frequencies in the frequency domain, only the most relevant frequency components are retained. This approach ensures that the model effectively retains the most important elements of the sequence and is better protected against noisy or redundant inputs, thus mitigating the problem of degradation of representations due to over-squeezing.

$$\mathrm{x_{i+1}} = \alpha \mathrm{f}(\mathrm{x_i}) + (1-\alpha)\mathcal{F}^{-1}(\mathcal{F}_{\mathrm{C}}(\mathrm{x_i}) \odot \mathrm{K}) \tag{13}$$

Where $\mathcal{F}$ denotes the Fourier transform, C denotes the threshold for frequency domain truncation, and α denotes the mixing coefficient.

Prediction Layer. After passing through L layers of blocks, we convert the output into normalized recommendation probabilities :

$$\hat{\mathrm{y}} = \mathrm{softmax}\left(\mathrm{E}^{\bullet}\ \ \mathrm{H}^{L}\right) \tag{14}$$

Therefore, we use the cross-entropy loss function to optimize the network.

$$\mathcal{L}_{\text{Rec}} = -\sum_{i=1}^{|I|}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{15}$$

Contrastive learning. We believe that contrastive learning can effectively promote the diversity of the embedding space, thereby alleviating the dimensional collapse problem. In PolyRec, we introduce contrastive learning strategy by passing the input $\mathrm{E_u}$ and $\mathrm{E_u'}$ through the encoder twice to obtain two outputs $\mathrm{H_u^L}$ and $(\mathrm{H_u^L})'$. By incorporating supervised augmented views, we optimize the item embedding distribution and alleviated the over-smoothing problem.

$$\mathcal{L}_{\mathrm{CLReg}(\mathbf{h}'_u,\mathbf{h}'_{u,s})} = -\log \frac{\exp\left(\mathrm{sim}\left(\mathrm{h}'_u,\mathrm{h}'_{u,s}\right)\right)}{\sum_{\mathrm{neg}} \exp\left(\mathrm{sim}\left(\mathrm{h}'_u,\mathrm{h}_{\mathrm{neg}}\right)\right)} \tag{16}$$

Where $\mathrm{sim}(\cdot)$ is dot product and $(\mathrm{h_u^L})'$ represent supervised augmented views.

$$\mathbf{h}'_u = (\mathbf{H}_u^L)'[-1], \mathbf{h}'_{u,s} = \left(\mathbf{H}_{u,s}^L\right)'[-1] \tag{17}$$

The finial loss function of PolyRec is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{Rec}} + \lambda \mathcal{L}_{\mathrm{CLReg}} \tag{18}$$

λ is a balance coefficient.

4 Experiments

We will validate the effectiveness of PolyRec by addressing the following four questions:
RQ1: How does PolyRec perform compared to other sequential recommendation models?
RQ2: How do key components affect the performance of PolyRec?
RQ3: How does different hyperparameters affect PolyRec?
RQ4: How to determine the order of the polynomial attention matrix?

4.1 Experimental Setup

Dataset. The summary information of the processed dataset is shown in the **Table 1**. The Amazon dataset consists of user reviews and metadata for various products available on Amazon. The dataset is divided into multiple categories, such as Beauty, Clothing, and Sports, allowing for domain-specific analysis and testing. Specifically, MovieLens-1M is an anonymized movie rating dataset that contains over 1 million interactions, making it highly suitable for validating models on large-scale data experiments.

Table 1. Datasets.

Datasets	Beauty	Clothing	Sports	ML-1M
Users	22,363	39,387	35,598	6,041
Items	12,101	23,033	18,357	3,417
Avg.Length	8.9	7.1	8.3	165.5
Actions	198,502	278,677	296,337	999,611
Sparsity	99.93%	99.97%	99.95%	95.16%

Baselines. We compare our model with representative models. Transformer based sequential recommendation includes SASRec[12], Cl4sRec[24], CoseRec[13] and DuoRec[15]. SASRec was the first to apply the Transformer model into sequential recommendation. CL4SRe creates various contrast views of the same user interaction sequence to facilitate contrast learning. CoseRec leverages item correlations to enhance data augmentation effectiveness within a contrast framework. DuoRec uses unsupervised model-level augmentation along with supervised semantic positive samples to facilitate contrast learning. FMLP is a full MLP model that uses a learnable filter enhancement module to remove noise from the embedding matrix. FEARec[7] take a frequency domain, using a ramp attention mechanism[17] to improve the original time-domain self-attention. MRGSRec[1] combines graph and sequential embeddings via a fusion layer. GSAU[3] use LightGCN as the graph encoder, SASRec as the sequential encoder whike sharing a unified embedding space optimized jointly.

Experiment Details. We replicate each baseline model using the optimal hyperparameters from the original paper. The PolyRec model is implemented in PyTorch. The hardware environment is an i5-12400F CPU and a RTX 3090 GPU.

4.2 Overall Performance Comparison (RQ1)

Table 2. Statistics of the datasets.

Datasets	Metric-Rec	SAS	BERT4	FMLP	CL4S	CoSe	Duo	FEA	MRGS	GSAU	Poly	Improve
Beauty	HR5	0.0365	0.0193	0.0398	0.0401	0.0537	0.0546	0.0597	0.0441	0.0525	**0.0602**	0.84%
	HR10	0.0627	0.0401	0.0632	0.0683	0.0752	0.0845	0.0884	0.0681	0.0868	**0.0934**	5.66%
	NDCG5	0.0236	0.0187	0.0258	0.0223	0.0361	0.0352	0.0366	0.0301	0.0336	**0.0375**	2.46%
	NDCG10	0.0281	0.0254	0.0333	0.0317	0.0430	0.0443	0.0459	0.0349	0.0447	**0.0483**	5.23%
Clothing	HR5	0.0168	0.0125	0.0173	0.0168	0.0175	0.0193	0.0214	0.0221	0.0208	**0.0231**	4.52%
	HR10	0.0272	0.0208	0.0277	0.0266	0.0279	0.0302	0.0323	0.0352	0.0335	**0.0355**	0.85%
	NDCG5	0.0091	0.0075	0.0098	0.0090	0.0095	0.0113	0.0121	0.0121	0.0122	**0.0130**	6.55%
	NDCG10	0.0124	0.0102	0.0127	0.0121	0.0131	0.0148	0.0156	0.0162	0.0163	**0.0170**	4.29%
Sports	HR5	0.0218	0.0176	0.0218	0.0227	0.0287	0.0326	0.0353	0.0289	0.0315	**0.0375**	6.23%
	HR10	0.0336	0.0326	0.0344	0.0374	0.0437	0.0498	0.0547	0.0434	0.0506	**0.0564**	3.11%
	NDCG5	0.0127	0.0105	0.0144	0.0129	0.0196	0.0208	0.0216	0.0186	0.0198	**0.0220**	1.85%
	NDCG10	0.0169	0.0153	0.0185	0.0184	0.0242	0.0262	0.0272	0.0227	0.0259	**0.0281**	3.31%
ML-1M	HR5	0.1087	0.0733	0.1109	0.1147	0.1262	0.2038	0.2212	0.1312	0.1662	**0.2230**	0.81%
	HR10	0.1904	0.1323	0.1932	0.1975	0.2212	0.2946	0.3123	0.2248	0.2482	**0.3172**	1.57%
	NDCG5	0.0638	0.0432	0.0657	0.0662	0.0761	0.1390	0.1523	0.0829	0.1078	**0.1577**	3.55%
	NDCG10	0.0910	0.0619	0.0918	0.0928	0.1021	0.1680	0.1861	0.1134	0.1342	**0.1882**	1.13%

In Table 2, we present the primary experimental results for PolyRec and all baseline moels on four datasets for HR and NDCG. To optimize the layout, we omitted the '-Rec' suffix from all models. As shown in the Table 2, PolyRec achieves a significant improvement compared to sequential recommendation baseline methods based on Transformer and contrastive learning. On the Sports dataset, PolyRec achieves the highest improvement of up to 6.5%.

4.3 Ablation Study (RQ2)

We sequentially remove each key component of the model to validate its effectiveness. In the Table, CL represents contrastive learning, TRC stands for truncated residual connections, PA denotes polynomial attention.

Table 3. Ablation Experiments on Four Datasets.

Methods	Beauty		Sports		Clothing		ML-1M	
	HR5	NDCG5	HR5	NDCG5	HR5	NDCG5	HR5	NDCG5
PolyRec	0.0602	0.0375	0.0375	0.0220	0.0231	0.0130	0.2230	0.1577
(a) w/o CL	0.0581	0.0356	0.0351	0.0197	0.0194	0.0106	0.2210	0.1540
(b) w/o TRC	0.0578	0.0359	0.0335	0.0190	0.0205	0.0116	0.2189	0.1499
(c) w/o PA	0.0591	0.0366	0.0356	0.0213	0.0209	0.0120	0.2199	0.1492

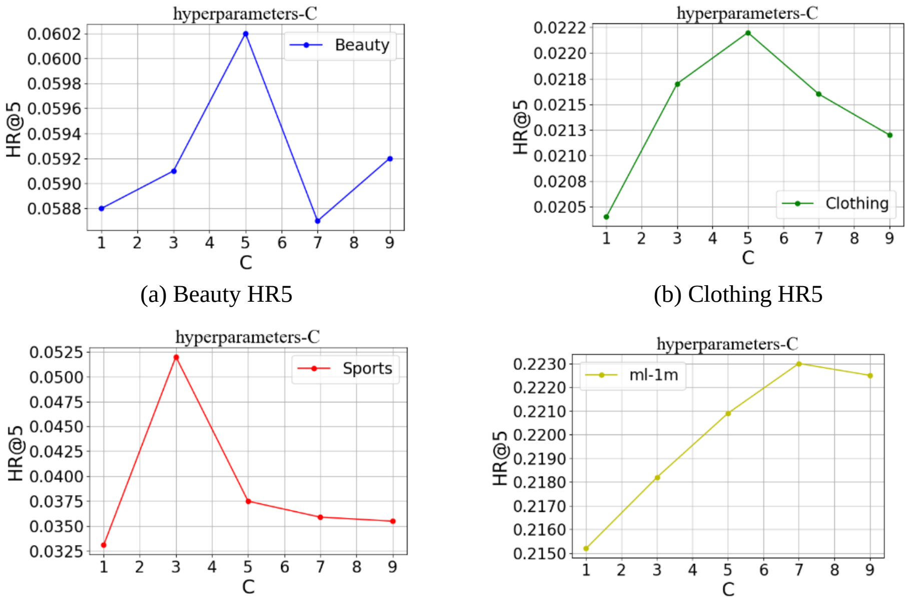
Truncated Residual Connections. As shown in **Table 3**, after removing the truncated residual connections, the model's performance declined to varying degrees across different datasets. Standard residual connections indiscriminately add full-spectrum information to the model output. While returning important information, noise and redundant information can excessively compress the effective information. Truncated residual connections address this by performing spectral truncation in the frequency domain, removing noise and excessive redundant information.

Polynomial Attention. Although truncated residual connections can transmit more important information from the input to the subsequent layers, thereby better preserving input information and maintaining the stability and integrity of signal flow within the network, a single attention matrix still suffers from representation degradation. As shown in **Table 3**, layers based on polynomial attention, through high-order terms and Chebyshev coefficients, can effectively improve the representational capability of the original attention mechanism.

Contrastive Learning. In traditional self-supervised learning, dimensional collapse typically refers to the phenomenon where the feature representations learned by the model become overly concentrated, resulting in the inability to effectively distinguish between most of the information in the feature space, which in turn impacts the model's performance. As shown in **Table 3**, Contrastive Learning effectively alleviates the dimensional collapse problem by enhancing the diversity of the embedding space.

4.4 Hyper-parameter Sensitivity (RQ3)

Truncation Coefficient C. In our experiments, we selected the truncation coefficient C as the threshold for truncating the user frequency spectrum in the frequency domain. Selecting an appropriate truncation threshold can more effectively filter out noise and preserve important components in the spectrum. As shown in **Fig. 4**, we test C values ranging from [1, 3, 5, 7, 9] and found that when C was selected between 3 and 5, the spectrum was effectively truncated, achieving optimal results.



(a) Beauty HR5

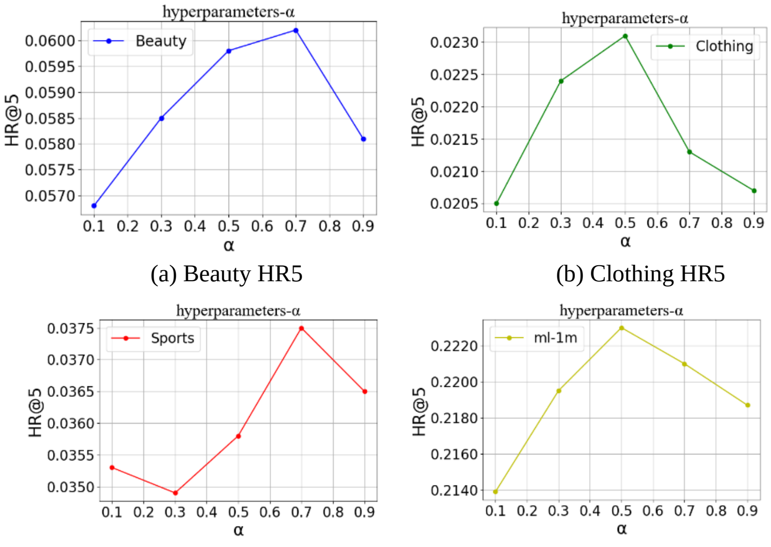
(b) Clothing HR5

(c)Sports HR5

(d)ml-1m HR5

Fig. 4. Truncation Coefficient C.

Mixing Coefficient α. In our experiments, we used the hyperparameter α to balance truncated residual connections and standard residual connections. As shown in **Fig. 5**, we tested α values ranging from [0.1, 0.3, 0.5, 0.7, 0.9], and found that optimal results were achieved when α was selected between 0.5 and 0.7.



(a) Beauty HR5

(b) Clothing HR5

(c)Sports HR5

(d)ml-1m HR5

Fig. 5. Mixing coefficient α.

4.5 Convergence Experiment (RQ4)

To balance model performance and computational complexity, it is necessary to select an appropriate order. To confirm the order of the polynomial matrix, this section incorporates PageRank theory to perform corresponding analysis and proofs. The attention matrix satisfies stochasticity, irreducibility, and aperiodicity. Therefore, we can treat the attention matrix as the P matrix in PageRank. According to the Perron-Frobenius theorem, higher-order terms must possess a stationary distribution, meaning the polynomial matrix is guaranteed to converge. As shown in the **Fig. 6** below, performing successive order-by-order differencing on the polynomial terms causes the color to become progressively lighter, indicating that higher-order terms are gradually dominated by lower-order terms.

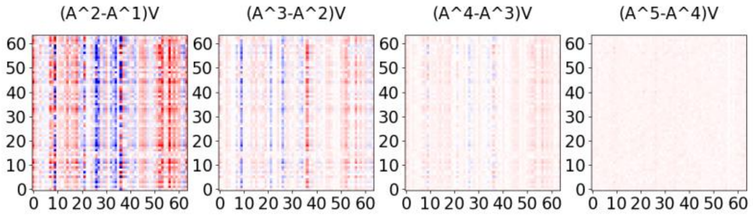


Fig. 6. Convergence Visualization. By performing successive differencing on the polynomial by order, it can be observed that higher-order terms are gradually absorbed by lower-order terms.

5 Conclusions and Prospect

This paper reviews the dimensional collapse issue in Transformer-based sequential recommendation models. Specifically, embedding vectors collapse into a low-dimensional subspace instead of covering the entire embedding space. This representation collapse causes the attention to become a low-rank matrix, making it difficult to represent users' preferences effectively. To mitigate this issue, We enhance two critical structures in Transformers. First, by introducing Fourier Transform, we truncate traditional residual connections in the frequency domain, enabling the transmission of more effective information in deep networks. Second, inspired by graph signal processing, we incorporate polynomial filters to extend the original attention layers, thereby strengthening the representational capacity of the attention mechanism. These improvements optimize the information flow within Transformers, alleviate over-squashing issues, and improve the distribution of embedding representations. In future work, we plan to explore the potential of Bernstein polynomials to further expand the capabilities of polynomial attention.

Acknowledgments. This work was supported by the National Science Foundation of China under Grant No.62467008, the Xinjiang Uygur Autonomous Region Outstanding Graduate Students Innovation Project No.XJDX2025YJS081, and the Xinjiang Uygur Autonomous Region Master's Student Innovation Project No.XJDX2025YJS179.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

[1]. Baikalov, V., Frolov, E.: End-to-end graph-sequential representation learning for accurate recommendations. In: Companion Proceedings of the ACM Web Conference 2024. pp. 501–504 (2024)

[2]. Bera, A., Wharton, Z., Liu, Y., Bessis, N., Behera, A.: Sr-gnn: Spatial relation-aware graph neural network for fine-grained image categorization. IEEE Transactions on Image Processing 31, 6017–6031 (2022)

[3]. Cao, Y., Yang, L., Liu, Z., Liu, Y., Wang, C., Liang, Y., Peng, H., Yu, P.S.: Graph-sequential alignment and uniformity: Toward enhanced recommendation systems. Companion Proceedings of the ACM Web Conference (2025)

[4]. Chen, J., Gao, K., Li, G., He, K.: Nagphormer: A tokenized graph transformer for node classification in large graphs. arXiv preprint arXiv:2206.04910 (2022)

[5]. Chen, J., Li, G., Hopcroft, J.E., He, K.: Signgt: Signed attention-based graph transformer for graph representation learning. arXiv preprint arXiv:2310.11025 (2023)

[6]. Dong, Y., Cordonnier, J.B., Loukas, A.: Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In: International Conference on Machine Learning. pp. 2793–2803. PMLR (2021)

[7]. Du, X., Yuan, H., Zhao, P., Qu, J., Zhuang, F., Liu, G., Liu, Y., Sheng, V.S.: Frequency enhanced hybrid attention network for sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 78–88 (2023)

[8]. He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 191–200. IEEE (2016)

[9]. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)

[10]. Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the eleventh ACM conference on recommender systems. pp. 306–310 (2017)

[11]. Jiang, J., Zhang, P., Luo, Y., Li, C., Kim, J.B., Zhang, K., Wang, S., Xie, X., Kim, S.: Adamct: adaptive mixture of cnn-transformer for sequential recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 976–986 (2023)

[12]. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM). pp. 197–206. IEEE (2018)

[13]. Liu, Z., Chen, Y., Li, J., Yu, P.S., McAuley, J., Xiong, C.: Contrastive self-supervised sequential recommendation with robust augmentation. arXiv preprint arXiv:2108.06479 (2021)

[14]. Noci, L., Anagnostidis, S., Biggio, L., Orvieto, A., Singh, S.P., Lucchi, A.: Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. Advances in Neural Information Processing Systems 35, 27198–27211 (2022)

[15]. Qiu, R., Huang, Z., Yin, H., Wang, Z.: Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 813–823 (2022)

[16]. Shin, Y., Choi, J., Wi, H., Park, N.: An attentive inductive bias for sequential recommendation beyond the self-attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 8984–8992 (2024)

[17]. Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., Yan, S.: Inception transformer. Advances in Neural Information Processing Systems 35, 23495–23509 (2022)

[18]. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 1441–1450 (2019)

[19]. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the eleventh ACM international conference on web search and data mining. pp. 565–573 (2018)

[20]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.:Attention is all you need. Advances in neural information processing systems 30 (2017)

[21]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.:Attention is all you need (2023), https://arxiv.org/abs/1706.03762

[22]. Wu, Q., Zhao, W., Yang, C., Zhang, H., Nie, F., Jiang, H., Bian, Y., Yan, J.: Simplifying and empowering transformers for large-graph representations. Advances in Neural Information Processing Systems 36(2024)

[23]. Wu, X., Ajorlou, A., Wu, Z., Jadbabaie, A.: Demystifying oversmoothing in attention-based graph neural networks. Advances in Neural Information Processing Systems 36 (2024)

[24]. Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B., Cui, B.: Contrastive learning for sequential recommendation. In: 2022 IEEE 38th international conference on data engineering (ICDE). pp. 1259–1273.IEEE (2022)

[25]. Zhou, K., Wang, H., Zhao, W.X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., Wen, J.R.: S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 1893–1902 (2020)