World Wide Web (2023) 26:3945–3964
https://doi.org/10.1007/s11280-023-01210-x

Graph neural network for recommendation in complex and quaternion spaces

Longcan Wu[1] · Daling Wang[1] · Shi Feng[1] · Xiangmin Zhou[2] · Yifei Zhang[1] · Ge Yu[1]

Received: 6 April 2023 / Revised: 29 August 2023 / Accepted: 30 August 2023 /
Published online: 10 October 2023

Abstract
With the development of graph neural network, researchers begin to use bipartite graph to model user-item interactions for recommendation. It is worth noting that most of graph recommendation models represent users and items in the real-valued space, which ignore the rich representational capacity of the non-real space. Besides, the simplicity and symmetry of the inner product make it ineffectively capture the intricate antisymmetric relations between users and items in interaction modeling. In this paper, based on the framework of graph neural network, we propose **G**raph **C**ollaborative **F**iltering for recommendation in **C**omplex and **Q**uaternion space (**GCFC** and **GCFQ** respectively). Specifically, we first use complex embeddings or quaternion embeddings to initialize users and items. Then, the Hermitian product (for GCFC) or Hamilton product (for GCFQ) and embedding propagation layers are used to further enrich the embeddings of users and items. As such, we can obtain both latent inter-dependencies and intra-dependencies between components of users and items. Finally, we aggregate the embeddings of different propagation layers and use the Hermitian or Hamilton product with inner product to obtain the intricate antisymmetric relations between users and items. We have carried out extensive experiments on four real-world datasets to verify the effectiveness of GCFC and GCFQ.

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2022*
Guest Editor: Richard Chbeir, Helen Huang, Yannis Manolopoulos, Fabrizio Silvestri

✉ Daling Wang
wangdaling@cse.neu.edu.cn

Longcan Wu
longcanwu@gmail.com

Shi Feng
fengshi@cse.neu.edu.cn

Xiangmin Zhou
xiangmin.zhou@rmit.edu.au

Yifei Zhang
zhangyifei@cse.neu.edu.cn

Ge Yu
yuge@cse.neu.edu.cn

1 Northeastern University, Shenyang, China

2 RMIT University, Melbourne, Australia

Keywords Recommendation · Collaborative filtering · Graph neural network · Non-real space

1 Introduction

As an important means to solve information overload, the recommender system has been widely studied in industry and academia. Most of the current recommendation models are based on collaborative filtering (CF). CF points out that similar users are interested in similar items. Model-based CF methods often use inner product to model the similarity between users and items, so nonlinear relations between users and items cannot be captured. With the development of deep learning, nonlinear neural networks are introduced into model-based CF methods [1, 2], achieving a good performance in multiple fields, such as social recommendations [3], sequential recommendations [4] and click-through rate predictions [5].

The above model-based CF methods are basically divided into two parts: embedding representation and interaction modeling. The purpose of embedding is to randomly initialize users and items into low-dimensional real-valued representations. Then an interaction module is devised to reconstruct the historical interactions between users and items. We can see that the user-item interactions are only used for model training. Actually, the user-item interactions have rich high-order collaborative signals. If we can integrate the high-order collaborative signals into the embedding representations, we can get better users and items embeddings. Based on this, some studies [6, 7] constructed a bipartite graph by user-item interactions, and used graph-related methods to capture the high-order collaborative signals between users and items in embedding representation process.

Despite the effectiveness of above models, we argue that these models have the following limitations. Firstly, the above models are based on real-valued operation and representations. Compared with representations in non-real space, such as complex space and quaternion space, real-valued representations have less representation capacity. As the most common non-real spaces, complex and quaternion spaces have attracted the attention of scholars in recent years, and have been widely used in different fields [8, 9]. The complex number $C = r + a\mathbf{i}$ and the quaternion $Q = r + a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ contain a real component and multiple imaginary components, so they have a richer representational capacity than the real number.

Secondly, the above models tend to use nonlinear neural networks in the interaction modeling, because the simplicity of inner product makes it ineffective capture the complex relationships between users and items. However, some studies have shown that the simple inner product can better model user preferences than nonlinear neural networks [10–12]. Therefore, the interaction modeling based on inner product is still worth studying and exploring. In the complex space, we can make the components of complex vectors interact with each other explicitly through the Hermitian product [13] and inner product, which makes it similar to the multi-view representations but exceeds the multi-view representations [14]. Similarly, Hamilton product in the quaternion space have the same advantages. Therefore, compared with inner product in real space, Hermitian product and Hamilton product have better modeling capability while maintaining the simplicity.

Thirdly, in the general recommendation, the user-item interactions can be modeled as a bipartite graph, where node represents user or item, and edge represents the interaction between user and item [15, 16]. From the perspective of bipartite graph, we can find that users and items belong to different sets, and there is obvious antisymmetric relations between them [14]. Neither inner product nor nonlinear neural network can model the antisymmetric relations between users and items in interaction modeling in above models. The Hermitian product of complex space and Hamilton product of quaternion space are not commutative, which makes them have great potential to capture the asymmetry in the recommendation system.

Thanks to the excellent properties of complex and quaternion space and the advantage of using graph to model recommendation, in this paper, based on the framework of graph neural network, we propose **G**raph **C**ollaborative **F**iltering for recommendation in **C**omplex and **Q**uaternion space (**GCFC** and **GCFQ**). Specifically, we first use complex representations or quaternion representations to initialize users and items, which endow users and items representations with a richer representational capacity. Then, based on Hermitian product (for GCFC) or Hamilton product (for GCFQ) and embedding propagation layers, we can further enrich the embeddings of users and items. Benefiting from the Hermitian or Hamilton product, we can obtain both latent inter-dependencies and intra-dependencies between components of users and items. By embedding propagation layers, we can obtain high-order connectivities between users and items. Finally, in interaction modeling layer, we aggregate the embeddings of different propagation layers and use the Hermitian or Hamilton product with inner product to capture the intricate antisymmetric relations between users and items. We apply GCFC and GCFQ on four real datasets, and the experimental results clearly demonstrate the superiority and effectiveness of our proposed model. This paper is an extension of our previous work [17]. In this paper, more technical and implementation details, more datasets, baselines, ablation experiments and discussions are included.

In summary, we make the following contributions: (1) We propose to model recommendation in complex and quaternion spaces from the perspective of graph. This work expands the research of recommendation in non-real space. (2) We propose two novel graph neural network models for recommendation in complex and quaternion spaces, GCFC and GCFQ, respectively. Based on Hermitian or Hamilton product and embedding propagation layers, we enrich embeddings of users and items, and capture the intricate antisymmetric relations between users and items as well. (3) We conduct extensive experiments on four commonly used real-world datasets. Experiment results show that GCFC and GCFQ achieve better performance than state-of-art recommendation solutions.

The rest of this paper is organized as follows: Section 2 introduces model-based collaborative filtering methods, graph-based recommendation and application of complex and quaternion neural networks; Section 3 gives the necessary mathematical background about complex and quaternion algebra; Section 4 describes two proposed graph recommendation models GCFC and GCFQ in detail; Section 5 presents extensive experiments on four real-world datasets to verify the effectiveness of GCFC and GCFQ; Section 6 provides a summary and future direction of this work.

2 Related work

2.1 Model-based collaborative filtering methods

The model-based CF methods can directly train model parameters according to downstream tasks, so it achieves better performance. Matrix factorization (MF) [18] is the most representative model-based CF method, which maps the ID of users and items to real-valued embeddings, and then uses the inner product as the interaction function. In order to obtain richer embeddings, various side information have been introduced, such as visual content, textual content, social network and knowledge graph (KG) [12]. In order to capture the nonlinear relationship between users and items, nonlinear neural network is introduced into the interaction function [1, 2]. However, some studies [10–12] have shown that the simple inner product can better model recommendation system than nonlinear neural networks. But the symmetry and simplicity make inner product ineffectively capture the complex asymmetric relationship between users and items. Different from above researches, we use Hermitian or Hamilton product with inner product to model the interactions between users and items in complex and quaternion spaces respectively. Hermitian product and Hamilton product have better modeling capabilities to obtain intricate antisymmetric relations between users and items while maintaining the simplicity with inner product.

2.2 Graph-based recommendation

From the perspective of graph, the interactions between users and items in the recommendation can be seen as a bipartite graph. Early work used random walks on the bipartite graph to obtain high-order connectivities of users and items to improve the performance [19, 20]. With the development of graph neural networks (GNN) [21], more and more researches begin to use GNN in the recommendation field, including social recommendations [3], sequential recommendation [22] and CTR prediction [23]. The GCMC [6] performs an embedding propagation on the user-item graph, which does not capture the high-order connectivities. NGCF [24] conducts multiple embedding propagation on user-item graph and concatenates multiple representations as the final embedding. PinSAGE [7] uses GCN to process the item-item graph to implicitly obtain high-order connectivities. For more information about the application of GNN in recommendation, please refer to the review [25]. Although the above work has achieved promising performance, they are all running in the real space, ignoring the rich representational capacity of complex space and quaternion space. Therefore, we propose to model recommender systems in complex and quaternion spaces from the perspective of graph. This paper extends the research of recommender systems in non-real space.

2.3 Application of complex and quaternion neural networks

Due to its richer representational capacity, complex-valued deep neural network [26] has been applied in many domains, including signal processing, computer vision and so on. Yang et al. [27] proposed a complex transformer for sequence modeling. Wisdom et al. [28] proposed RNN in complex space and applied it in a real-world speech task. [13, 29] used complex-valued embeddings to model the knowledge graph to capture antisymmetric relationships. For more application of complex neural network, please refer to the review [9].

Since real-world data is often multidimensional, we need a specific approach to consider the relationship between different dimensions. Quaternion, as an extension of complex number, can consider up to four-dimensional information. Therefore, quaternion neural networks have attracted the attention of researchers. Parcollet et al. [30] used quaternion convolutional neural networks for image reconstruction. Shi et al. [31] proposed quaternion block network to learn multi-modality interaction for visual question answering. Nguyen et al. [32] employed graph neural network in quaternion space for node classification. Zhang et al. [33] used quaternion space to model entities and relationships in knowledge graph. Tay et al. [34] proposed quaternion transformer for many NLP tasks. For more application of quaternion neural network, please refer to the review [8].

As far as we know, only a few papers currently consider using complex number or quaternion to represent users and items in recommendations. Zhang et al. [14] directly employed complex embedding and quaternion embedding to model recommendation on the basis of MF. Fang et al. [35] modeled users and items in quaternion space and propagate them with quaternion feature transformation. Tran et al. [36] used attention and RNN to model user's long-term and short-term preferences for sequential recommendation in quaternion space. Inspired by the modeling of KG in quaternion space, Li et al. [37] proposed to model unified user-item KG in quaternion space for KG-aware recommendation. The above researches show that modeling recommendation in complex and quaternion spaces can achieve better performance. Different from the above studies, to the best of our knowledge, we are the first to use GNN to model user-item interaction in complex and quaternion spaces. Based on Hermitian or Hamilton product and embedding propagation layers, we can enrich embeddings of users and items and capture the intricate antisymmetric relations between users and items.

3 Background of complex and quaternion algebra

In this section, we give the necessary mathematical background about complex and quaternion algebra. For more details, please refer to [8, 9].

Complex algebra. A complex number C, belonging to complex space $\mathbb{C}$, contains a real part and an imaginary part: $C = r + a\mathbf{i}$, where r and a are real numbers and the imaginary unit $\mathbf{i}$ satisfies $\mathbf{i}^2 = -1$. We can expand the real and imaginary parts into real-valued vectors to obtain a complex vector. Similarly, we can obtain a complex matrix. The Hermitian product [13] of two complex number is defined as:

$$\langle C_1, C_2\rangle = \overline{C_1}C_2 = (r_1 - a_1\mathbf{i})(r_2 + a_2\mathbf{i}) = (r_1r_2 + a_1a_2) + (r_1a_2 - r_2a_1)\mathbf{i} \tag{1}$$

where $\overline{C_1} = r_1 - a_1\mathbf{i}$ represents the complex conjugate of C_1. From above formula, we can find the Hermitian product of two complex number is asymmetrical, that is $\langle C_1, C_2\rangle \neq \langle C_2, C_1\rangle$. Many operations in real space can be applied to complex number. Suppose f is an operator in the real number space, we can use f in two complex numbers as follows: $f(C_1, C_2) = f(r_1, r_2) + f(a_1, a_2)\mathbf{i}$.

Quaternion algebra. A quaternion Q is an extension of complex number, belonging to quaternion space $\mathbb{Q}$. Q contains one real part and three imaginary parts: $Q = r+a\mathbf{i}+b\mathbf{j}+c\mathbf{k}$, where r, a, b, and c are all real numbers and the imaginary units $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ satisfy: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$; $\mathbf{ij} = \mathbf{k}$, $\mathbf{jk} = \mathbf{i}$, $\mathbf{ki} = \mathbf{j}$, $\mathbf{ji} = -\mathbf{k}$, $\mathbf{kj} = -\mathbf{i}$, $\mathbf{ik} = -\mathbf{j}$. We can expand the real and imaginary parts into real-valued vectors to obtain a quaternion vector. Similarly, we can obtain a quaternion matrix. The Hamilton product of two quaternion is also quaternion:

$$\begin{aligned} Q_1 \otimes Q_2 &= (r_1 + a_1\mathbf{i} + b_1\mathbf{j} + c_1\mathbf{k}) \otimes (r_2 + a_2\mathbf{i} + b_2\mathbf{j} + c_2\mathbf{k}) \\ &= (r_1r_2 - a_1a_2 - b_1b_2 - c_1c_2) + (r_1a_2 + a_1r_2 + b_1c_2 - c_1b_2)\mathbf{i} \\ &+ (r_1b_2 - a_1c_2 + b_1r_2 + c_1a_2)\mathbf{j} + (r_1c_2 + a_1b_2 - b_1a_2 + c_1r_2)\mathbf{k} \end{aligned} \tag{2}$$

From above formula, we can find that the Hamilton product is not commutative, that is $Q_1 \otimes Q_2 \neq Q_2 \otimes Q_1$. Many operations in real space can be applied to quaternion. Suppose f is an operator in the real number space, we can use f in two quaternions as follows: $f(Q_1, Q_2) = f(r_1, r_2) + f(a_1, a_2)\mathbf{i} + f(b_1, b_2)\mathbf{j} + f(c_1, c_2)\mathbf{k}$.

4 Methodology

In this section, we will introduce the proposed GCFC and GCFQ models in detail. Before that, we first give the definition of the problem. Then we give an explanation of the model overall framework. Next, we will detail each part of models.

4.1 Problem formulation

Given user set $\mathcal{U} = \{u_1, u_2, ..., u_m\}$, item set $\mathcal{V} = \{v_1, v_2, v_3, ..., v_n\}$, and user-item interaction matrix $R \in \mathbb{R}^{m\times n}$, we can construct the user-item bipartite graph $G = (\{\mathcal{U}, \mathcal{V}\}, A)$. In user-item interaction matrix R, if there is implicit feedback between user u and item v, such as purchasing, clicking, watching, then $R_{uv} = 1$, otherwise $R_{uv} = 0$. $A \in \mathbb{R}^{(m+n)\times(m+n)}$ is the adjacency matrix of the user-item graph, which is constructed from user-item interaction matrix R:

$$A = \begin{bmatrix} 0^{(m\times m)} & R \\ R^T & 0^{(n\times n)} \end{bmatrix} \tag{3}$$

Our task is to learn the low-dimensional vector representations of users and items on the bipartite graph G, design the prediction functions to calculate the probabilities of each user engaging an item in complex and quaternion spaces, and make Top-K recommendations for a target user based on the probability scores.

4.2 Framework overview

Since GCFC and GCFQ have similar framework, we only give the overall framework of GCFQ as shown in Figure 1. The GCFQ is mainly composed of three parts: (1) In the embedding layer, we randomly initialize quaternion vectors as the embeddings of users and items. (2) In the embedding propagation layers, by continuously gathering information from neighbors, the high-order connectivity between users and items are integrated into the embeddings of users and items. Thanks to the Hamilton product, we can get both latent inter-dependencies and intra-dependencies between components of users and items. (3) In

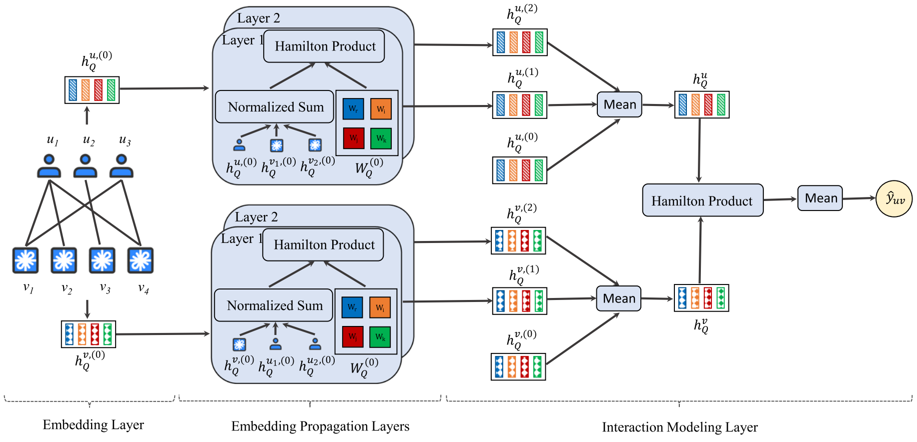
$h_Q^{u,(0)}$
u_1 u_2 u_3
v_1 v_2 v_3 v_4
$h_Q^{v,(0)}$
Layer 2
Layer 1
Hamilton Product
Normalized Sum
W_r W_i W_j W_k
$h_Q^{u,(0)}$ $h_Q^{v_1,(0)}$ $h_Q^{v_2,(0)}$ $W_Q^{(0)}$
$h_Q^{v,(0)}$ $h_Q^{u_1,(0)}$ $h_Q^{u_2,(0)}$ $W_Q^{(0)}$
$h_Q^{u,(2)}$ $h_Q^{u,(1)}$ $h_Q^{u,(0)}$
$h_Q^{v,(2)}$ $h_Q^{v,(1)}$ $h_Q^{v,(0)}$
Mean
h_Q^u h_Q^v
Hamilton Product
Mean
$\hat{y}_{uv}$
Embedding Layer
Embedding Propagation Layers
Interaction Modeling Layer

Figure 1 Overall architecture of the proposed model GCFQ

the interaction modeling layer, we aggregate the embeddings of different propagation layers as the final embeddings of users and items, and then use Hamilton product with inner product to calculate the affinity score between users and items.

4.3 GCFC

4.3.1 Embedding layer

In the embedding layer, we use complex vectors as embeddings of users and items. Note that we use real vectors to represent the real and imaginary part of complex vector, and then use real operations to simulate complex operations [27]. So, we have two initial complex embedding matrices $H_C^{U,(0)} \in \mathbb{C}^{m\times d}$ and $H_C^{V,(0)} \in \mathbb{C}^{n\times d}$, where m and n are the number of users and items respectively, d is the embedding size. Specifically, the initial complex embedding of user u is $h_C^{u,(0)} = h_{C,r}^{u,(0)} + h_{C,i}^{u,(0)}\mathbf{i}$, where $h_{C,r}^{u,(0)}, h_{C,i}^{u,(0)} \in \mathbb{R}^{d/2}$, $\mathbf{i}$ is an imaginary unit. Similarly, the initial complex embedding of item v is $h_C^{v,(0)} = h_{C,r}^{v,(0)} + h_{C,i}^{v,(0)}\mathbf{i}$, where $h_{C,r}^{v,(0)}, h_{C,i}^{v,(0)} \in \mathbb{R}^{d/2}$. Thanks to complex embeddings, the user and item representations have two parts with a richer representational capacity.

4.3.2 Embedding propagation layer

The user-item interaction graph contains the high-order collaborative signals [24, 38, 39]. The collaborative signals reflect the similarity between users and items. In order to integrate the high-order collaborative signals into the embeddings of users and items, with the help of message-passing architecture in GNN [21], we introduce embedding propagation in the user-item graph. Specifically, the embedding propagation layer includes two steps: feature propagation and nonlinear transformation.

Feature propagation. When user u performs feature propagation in l_{th} layer, u aggregates the message from graph neighbors N_u and itself embeddings in layer $l-1$. We take feature aggregation function in matrix form used in GCN [21]:

$$L = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}} \tag{4}$$

$$H_C^{(l)} = \langle L, H_C^{(l-1)}\rangle = LH_{C,r}^{(l-1)} + LH_{C,i}^{(l-1)}\mathbf{i} \tag{5}$$

where A is the adjacency matrix of the user-item graph G, $\tilde{A} = A + I$ and I is the identity matrix; $\widetilde{D}$ is the degree matrix of $\tilde{A}$; $L \in \mathbb{R}^{(m+n)\times(m+n)}$represents the normalized adjacency matrix; $\langle,\rangle$ denotes the Hermitian product; $H_C^{(l-1)} = H_{C,r}^{(l-1)} + H_{C,i}^{(l-1)}\mathbf{i}$ is the embeddings of users and items after $l-1$ step of embedding propagation and $H_C^{(l-1)} \in \mathbb{C}^{(m+n)\times d_{l-1}}$, $H_C^{(0)} = \left[H_C^{U,(0)}; H_C^{V,(0)}\right]$.

Nonlinear transformation. After obtaining information from neighbors, we perform feature transformation in formula (6) and nonlinear activation in formula (7):

$$\begin{aligned} H_C^{(l)} &= \langle H_C^{(l)}, W_C^{(l)}\rangle \\ &= \left(H_{C,r}^{(l)} - H_{C,i}^{(l)}\mathbf{i}\right)\left(W_{C,r}^{(l)} + W_{C,i}^{(l)}\mathbf{i}\right) \end{aligned}$$

$$= \left(H_{C,r}^{(l)} W_{C,r}^{(l)} + H_{C,i}^{(l)} W_{C,i}^{(l)}\right) + \left(H_{C,r}^{(l)} W_{C,i}^{(l)} - H_{C,i}^{(l)} W_{C,r}^{(l)}\right) \mathbf{i} \tag{6}$$

$$H_C^{(l)} = \sigma\left(H_C^{(l)}\right) = \sigma\left(H_{C,r}^{(l)}\right) + \sigma\left(H_{C,i}^{(l)}\right) \mathbf{i} \tag{7}$$

where $W_C^{(l)} = W_{C,r}^{(l)} + W_{C,i}^{(l)}\mathbf{i}$ is the trainable complex weight matrix and $W_C^{(l)} \in \mathbb{C}^{(d_{l-1}/2)\times d_l}$; $\sigma(x)$ is a nonlinear activation function. From above formula, we can find that $W_{C,r}^{(l)}$ and $W_{C,i}^{(l)}$ are shared by the real and an imaginary parts of $H_C^{(l)}$ in Hermitian product. Weight sharing enables the model to obtain both latent inter-dependencies and intra-dependencies between the components of user and item, leading to a higher expressive model.

4.3.3 Interaction modeling layer

After L layers, we will obtain $L + 1$ representations at different layers for user u and item v: $\left\{h_C^{u,(0)}, h_C^{u,(1)}, \ldots, h_C^{u,(L)}\right\}$, $\left\{h_C^{v,(0)}, h_C^{v,(1)}, \ldots, h_C^{v,(L)}\right\}$. We take the mean of $L + 1$ representations to get the final representation h_C^u and h_C^v for user and item. For example, the calculation formula for h_C^u is as follows:

$$\begin{aligned} h_C^u &= \text{mean}\left(h_C^{u,(0)}, h_C^{u,(1)}, \ldots h_C^{u,(L)}\right) \\ &= \text{mean}\left(h_{C,r}^{u,(0)}, \ldots, h_{C,r}^{u,(L)}\right) + \text{mean}\left(h_{C,i}^{u,(0)}, \ldots, h_{C,i}^{u,(L)}\right)\mathbf{i} \end{aligned} \tag{8}$$

Then we use the Hermitian product to model the interaction between users and items:

$$\begin{aligned} r_C &= \langle h_C^u, h_C^v \rangle \\ &= \left(h_{C,r}^u - h_{C,i}^u \mathbf{i}\right)\left(h_{C,r}^v + h_{C,i}^v \mathbf{i}\right) \\ &= \left(h_{C,r}^u \cdot h_{C,r}^v + h_{C,i}^u \cdot h_{C,i}^v\right) + \left(h_{C,r}^u \cdot h_{C,i}^v - h_{C,i}^u \cdot h_{C,r}^v\right)\mathbf{i} \end{aligned} \tag{9}$$

where "$\cdot$" denotes dot product. Through Hermitian product, we can further obtain inter-dependencies between components of user and item. In addition, Hermitian product is asymmetrical, which well captures the asymmetric relationship between user and item. Finally we take mean of all components of r_C as the prediction score:

$$\hat{y}_{uv} = \left(r_{C,r} + r_{C,i}\right)/2 \tag{10}$$

4.4 GCFQ

4.4.1 Embedding layer

In the embedding layer, we use quaternion vectors as the embeddings of users and items. Thus, we have two initial quaternion embedding matrices $H_Q^{U,(0)} \in \mathbb{Q}^{m\times d}$ and $H_Q^{V,(0)} \in \mathbb{Q}^{n\times d}$. Specifically, the initial quaternion embedding of user u is $h_Q^{u,(0)} = h_{Q,r}^{u,(0)} + h_{Q,i}^{u,(0)}\mathbf{i} + h_{Q,j}^{u,(0)}\mathbf{j} + h_{Q,k}^{u,(0)}\mathbf{k}$, where $h_{Q,r}^{u,(0)}, h_{Q,i}^{u,(0)}, h_{Q,j}^{u,(0)}, h_{Q,k}^{u,(0)} \in \mathbb{R}^{d/4}$, $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary units. Similarly, the initial quaternion embedding of item v is $h_Q^{v,(0)} = h_{Q,r}^{v,(0)} + h_{Q,i}^{v,(0)}\mathbf{i} + h_{Q,j}^{v,(0)}\mathbf{j} + h_{Q,k}^{v,(0)}\mathbf{k}$, where $h_{Q,r}^{v,(0)}, h_{Q,i}^{v,(0)}, h_{Q,j}^{v,(0)}, h_{Q,k}^{v,(0)} \in \mathbb{R}^{d/4}$. Using quaternion embeddings, the user and item representations have four parts with a richer representational capacity.

4.4.2 Embedding propagation layer

References to GCFC model above, we provide the matrix form of the embedding propagation layer in GCFQ. The embedding propagation consists of two steps: feature propagation and nonlinear transformation.

Feature propagation. We take feature aggregation function in matrix form used in GCN [21]:

$$H_Q^{(l)} = L \otimes H_Q^{(l-1)} = LH_{Q,r}^{(l-1)} + LH_{Q,i}^{(l-1)}\mathbf{i} + LH_{Q,j}^{(l-1)}\mathbf{j} + LH_{Q,k}^{(l-1)}\mathbf{k} \tag{11}$$

where L represents the normalized adjacency matrix; $\otimes$ denotes the Hamilton product; $H_Q^{(l-1)} = H_{Q,r}^{(l-1)} + H_{Q,i}^{(l-1)}\mathbf{i} + H_{Q,j}^{(l-1)}\mathbf{j} + H_{Q,k}^{(l-1)}\mathbf{k}$ is the embedding of users and items after $l-1$ step of embedding propagation and $H_Q^{(l-1)} \in \mathbb{Q}^{(m+n)\times d_{l-1}}$, $H_Q^{(0)} = \left[H_Q^{U,(0)}; H_Q^{V,(0)}\right]$.

Nonlinear transformation. After obtaining the information from neighbors, we perform feature transformation in formula (12) and nonlinear activation in formula (13):

$$\begin{aligned}
H_Q^{(l)} &= H_Q^{(l)} \otimes W_Q^{(l)} \\
&= \left(H_{Q,r}^{(l)}W_{Q,r}^{(l)} - H_{Q,i}^{(l)}W_{Q,i}^{(l)} - H_{Q,j}^{(l)}W_{Q,j}^{(l)} - H_{Q,k}^{(l)}W_{Q,k}^{(l)}\right) \\
&+ \left(H_{Q,r}^{(l)}W_{Q,i}^{(l)} + H_{Q,i}^{(l)}W_{Q,r}^{(l)} + H_{Q,j}^{(l)}W_{Q,k}^{(l)} - H_{Q,k}^{(l)}W_{Q,j}^{(l)}\right)\mathbf{i} \\
&+ \left(H_{Q,r}^{(l)}W_{Q,j}^{(l)} - H_{Q,i}^{(l)}W_{Q,k}^{(l)} + H_{Q,j}^{(l)}W_{Q,r}^{(l)} + H_{Q,k}^{(l)}W_{Q,i}^{(l)}\right)\mathbf{j} \\
&+ \left(H_{Q,r}^{(l)}W_{Q,k}^{(l)} + H_{Q,i}^{(l)}W_{Q,j}^{(l)} - H_{Q,j}^{(l)}W_{Q,i}^{(l)} + H_{Q,k}^{(l)}W_{Q,r}^{(l)}\right)\mathbf{k}
\end{aligned} \tag{12}$$

$$H_Q^{(l)} = \sigma\left(H_Q^{(l)}\right) = \sigma\left(H_{Q,r}^{(l)}\right) + \sigma\left(H_{Q,i}^{(l)}\right)\mathbf{i} + \sigma\left(H_{Q,j}^{(l)}\right)\mathbf{j} + \sigma\left(H_{Q,k}^{(l)}\right)\mathbf{k} \tag{13}$$

where $W_Q^{(l)} = W_{Q,r}^{(l)} + W_{Q,i}^{(l)}\mathbf{i} + W_{Q,j}^{(l)}\mathbf{j} + W_{Q,k}^{(l)}\mathbf{k}$ is the trainable quaternion weight matrix and $W_Q^{(l)} \in \mathbb{Q}^{(d_{l-1}/4)\times d_l}$; $\sigma(x)$ is a nonlinear activation function. From above formulas, we can find that $W_{Q,r}^{(l)}$, $W_{Q,i}^{(l)}$, $W_{Q,j}^{(l)}$, $W_{Q,k}^{(l)}$ are shared by the real and three imaginary parts of $H_Q^{(l)}$ in Hamilton product. Weight sharing enables the model to obtain both latent inter-dependencies and intra-dependencies between components of user and item, leading to a higher expressive model.

4.4.3 Interaction modeling layer

After L layers, we will obtain $L+1$ representations at different layers for user u and item v: $\left\{h_Q^{u,(0)}, h_Q^{u,(1)}, \ldots, h_Q^{u,(L)}\right\}$, $\left\{h_Q^{v,(0)}, h_Q^{v,(1)}, \ldots, h_Q^{v,(L)}\right\}$. We take the mean of $L+1$ representations to get the final representation h_Q^u and h_Q^v for user and item. Then we use the Hamilton product to model the interaction between users and items:

$$\begin{aligned}
r_Q &= h_Q^u \otimes h_Q^v \\
&= \left(h_{Q,r}^u \cdot h_{Q,r}^v - h_{Q,i}^u \cdot h_{Q,i}^v - h_{Q,j}^u \cdot h_{Q,j}^v - h_{Q,k}^u \cdot h_{Q,k}^v\right) \\
&+ \left(h_{Q,r}^u . h_{Q,i}^v + h_{Q,i}^u \cdot h_{Q,r}^v + h_{Q,j}^u \cdot h_{Q,k}^v - h_{Q,k}^u \cdot h_{Q,j}^v\right)\mathbf{i}
\end{aligned}$$

$$+\left(h^{u}_{Q,r}.h^{v}_{Q,j}-h^{u}_{Q,i}\cdot h^{v}_{Q,k}+h^{u}_{Q,j}\cdot h^{v}_{Q,r}+h^{u}_{Q,k}\cdot h^{v}_{Q,i}\right)\mathbf{j}$$
$$+\left(h^{u}_{Q,r}.h^{v}_{Q,k}+h^{u}_{Q,i}\cdot h^{v}_{Q,j}-h^{u}_{Q,j}\cdot h^{v}_{Q,i}+h^{u}_{Q,k}\cdot h^{v}_{Q,r}\right)\mathbf{k} \quad (14)$$

where "·" denotes dot product. Through Hamilton product, we can further obtain the interdependencies between the components of user and item. In addition, Hamilton product is asymmetrical, which well captures the asymmetric relationship between user and item. Finally we take mean of all components of r_Q as the prediction score:

$$\hat{y}_{ui}=\left(r_{Q,r}+r_{Q,i}+r_{Q,j}+r_{Q,k}\right)/4 \quad (15)$$

4.5 Model training

This work mainly focuses on the top-K recommendation task, and we optimize the model using Bayesian Personalized Ranking (BPR) loss function [18]. Specifically, BPR loss function is formulated as:

$$Loss=\sum_{(u,i,j)\in T}-\ln\sigma\left(\hat{y}_{ui}-\hat{y}_{uj}\right)+\lambda\|\Theta\|_2^2 \quad (16)$$

where $T=\{(u,i,j)\mid(u,i)\in\mathcal{R}_+,(u,j)\in\mathcal{R}_-\}$ is training dataset with observed interactions set $\mathcal{R}_+$ and the unobserved interactions set $\mathcal{R}_-$; $\sigma(x)$ is sigmoid function; $\hat{y}_{ui}$ and $\hat{y}_{uj}$ are the learned prediction score; λ is the L_2 regularization coefficient; Θ denotes trainable parameters, including $H^{U,(0)}_{C/Q}$, $H^{V,(0)}_{C/Q}$, $W^{(l)}_{C/Q}$.

5 Experiments

In this section, we conduct extensive experiments over four real-world datasets to verify the effectiveness of proposed models GCFC and GCFQ. We first give detailed experimental settings. Then we give model performance, hyper-parameter studies and model analysis.

5.1 Experimental settings

5.1.1 Datasets

We conduct experiments on four widely used benchmark datasets [12], including Amazon-Cloth, Amazon-Music, Amazon-Electronic and Book-Crossing. Information about these datasets is shown in Table 1. Amazon-Cloth, Amazon-Music and Amazon-Electronic are datasets that users rate items on Amazon, respectively corresponding to three categories: clothing, digital music and electronics. For each user, we take score greater than 3 as positive feedback. For the Amazon datasets, we respectively use 5-core setting, 10-core setting and 5-core setting [24] to ensure the quality of the data. Book-Crossing is the dataset of user ratings about books. For each user, we take score greater than 0 as positive feedback and use 10-core setting to ensure the quality of data.

Table 1 Datasets statistics

Dataset	#User	#Item	#Interactions	Density
Amazon-Cloth	4,810	3,368	31,122	0.0019
Amazon-Music	3,765	2,663	43,190	0.0043
Amazon-Electronic	9,279	6,065	158,979	0.0028
Book-Crossing	6,754	13,670	374,325	0.0040

5.1.2 Baselines

We choose some representative baselines to perform comparison experiments. These models not only include MF, MLP, NeuMF, DMF, GCMC, PinSage, NGCF in real space, but also CCF and QCF in non-real space. Since this work focuses on general recommendation, we do not use sequential recommendation model [36] and KG aware recommendation model [37] in non-real space as baselines. The brief introduction to the above-mentioned baselines is as follows:

- **MF** [18]: This model randomly initializes embeddings of users and items, and then uses the inner product to predict the affinity score.
- **MLP** and **NeuMF** [1]: MLP uses nonlinear neural network to model interaction. Based on MF and MLP, NeuMF uses deep neural networks and inner product jointly to predict the affinity score between user and item.
- **DMF** [2]: This model uses ratings as feature of user and item, maps the feature by nonlinear neural network, and predicts the affinity score between user and item by cosine similarity.
- **GCMC** [6]: This model uses graph auto-encoder to generate user and item embeddings for recommendation.
- **PinSage** [7]: This model uses GraphSAGE to generate item embeddings on the item-item graph. In this paper, we directly use GraphSAGE on user-item interaction graph for recommendation task.
- **NGCF** [24]: This model proposes to perform multiple embedding propagation on the user-item interaction graph to capture the collaborative signals between users and items.
- **CCF** and **QCF** [14]: These two models use complex representation and quaternion representation as the embeddings of users and items on the basis of MF.

5.1.3 Parameter settings and evaluation setup

For all baselines, batch size is set to 1024, embedding dimensionality is set to 64, adam optimizer [40] is used to optimize all models, Xavier initializer is used to initialize the model parameters, and the loss function is BPR loss. For other hyperparameters, we refer to the original paper of baselines. Note that, the dimensionality of complex embeddings of users and items is 64 for GCFC, which means that each component of complex embedding is a vector with size $64/2 = 32$ as we mentioned in Section 4.3. Similarly, each component of quaternion embedding is a vector with size $64/4 = 16$ for GCFQ. In order to obtain high-order connectivity between users and items, the number of embedding propagation layer of GCFC and GCFQ is searched in $\{1, 2, 3, 4\}$. Finally, for all baselines, Bayesian HyperOpt [12] is used to perform hyper-parameter optimization on learning rate and coefficients of L_2 regularization term w.r.t. NDCG@20 on each dataset for 30 trails.

In the experiment, we randomly select 80% of the historical interactions between users and items as the training set, 10% as the validation set, and 10% as the test set. All models are trained on the training set, and the optimal model parameters are obtained on the validation set. The evaluation metrics use the two most common ranking metrics Recall@K and NDCG@K by the all-ranking protocol [38]. The above process is executed 10 times and the final average result is presented.

5.2 Overall performance comparison

The comparison results of all models are shown in Tables 2 and 3 with the best result hightlighted in bold. We can draw the following conclusions:

- GCFQ performs best in all datasets, which shows the effectiveness of using graph neural network to model recommendation in quaternion space. Compared with GCFC, GCFQ uses quaternion embeddings, which have four components and enhance the representational capacity of model. Thus, GCFQ performs better than GCFC. On Amazon-Cloth and Amazon-Electronic datasets, GCFC gets similar performance to GCMC, PinSage and NGCF. On Amazon-Music and Book-Crossing datasets, GCFC achieves the best performance except for GCFQ, which indicates that the GCFC is more suitable for dataset with higher density and shows the potential of modeling recommendation in complex space.
- Compared with the models MF, MLP, NeuMF and DMF in real space, the models (CCF, QCF, GCFC, GCFQ) in the non-real space can achieve better performance. This is because the non-real embeddings have the rich representational capacity and Hamilton product and Hermitian product can capture the intricate antisymmetric relation between users and items. In addition, in most cases, MF, CCF and QCF performe better than MLP and NeuMF models using nonlinear neural networks, demonstrating the effectiveness of using inner product to model interaction between users and items, which is consistent with previous studies [10–12].
- Compared with MF, CCF and QCF, the models that use bipartite graphs to model the interactions between users and items (NGCF, GCFC, GCFQ, etc.) can achieve better

Table 2 Performance of all methods on Amazon-Cloth and Amazon-Music datasets

	Amazon-Cloth				Amazon-Music			
	Recall@K		NDCG@K		Recall@K		NDCG@K	
Model	K=20	K=30	K=20	K=30	K=20	K=30	K=20	K=30
MF	0.0760	0.0903	0.0327	0.0358	0.1352	0.1718	0.0592	0.0681
MLP	0.0324	0.0404	0.0152	0.0169	0.1141	0.1502	0.0495	0.0581
NeuMF	0.0384	0.0466	0.0153	0.0171	0.0885	0.1184	0.0408	0.0480
DMF	0.0729	0.0861	0.0311	0.0339	0.1383	0.1712	0.0581	0.0674
GCMC	0.0686	0.0883	0.0321	0.0363	0.1388	0.1770	0.0591	0.0682
PinSage	0.0709	0.0878	0.0356	0.0393	0.1429	0.1801	0.0647	0.0736
NGCF	0.0878	0.1021	0.0363	0.0390	0.1402	0.1760	0.0635	0.0722
CCF	0.0702	0.0896	0.0298	0.0339	0.1442	0.1816	0.0647	0.0737
QCF	0.0874	0.1058	0.0397	0.0436	0.1449	0.1868	0.0658	0.0749
GCFC	0.0815	0.0962	0.0393	0.0425	0.1541	0.1934	0.0666	0.0759
GCFQ	**0.0881**	**0.1074**	**0.0405**	**0.0437**	**0.1553**	**0.1973**	**0.0678**	**0.0778**

Table 3 Performance of all methods on Amazon-Electronic and Book-Crossing datasets

	Amazon-Electronic				Book-Crossing			
	Recall@K		NDCG@K		Recall@K		NDCG@K	
Model	K=20	K=30	K=20	K=30	K=20	K=30	K=20	K=30
MF	0.0301	0.0387	0.0125	0.0144	0.0020	0.0024	0.0009	0.0010
MLP	0.0276	0.0378	0.0113	0.0136	0.0021	0.0024	0.0010	0.0011
NeuMF	0.0339	0.0444	0.0142	0.0165	0.0017	0.0021	0.0008	0.0009
DMF	0.0282	0.0348	0.0138	0.0153	0.0024	0.0030	0.0012	0.0013
GCMC	0.0423	0.0559	0.0184	0.0214	0.0027	0.0033	0.0014	0.0015
PinSage	0.0458	0.0587	0.0197	0.0225	0.0025	0.0030	0.0014	0.0015
NGCF	0.0436	0.0555	0.0186	0.0212	0.0024	0.0030	0.0013	0.0014
CCF	0.0309	0.0402	0.0143	0.0164	0.0026	0.0031	0.0011	0.0013
QCF	0.0455	0.0578	0.0197	0.0225	0.0028	0.0034	0.0014	0.0015
GCFC	0.0450	0.0581	0.0193	0.0222	0.0033	0.0039	0.0016	0.0018
GCFQ	**0.0471**	**0.0592**	**0.0205**	**0.0233**	**0.0034**	**0.0041**	**0.0018**	**0.0020**

performance. The user-item interaction graph contains the high-order collaborative signals, which reflect the similarity between users and items. By performing message-passing on the bipartite graph, the high-order collaborative signals can be integrated into the embedings of users and items.

- We attribute the excellent performance of GCFQ (GCFC) to the following reasons: (1) thanks to the quaternion (complex) representations, the embeddings of user and item have a richer representational capacity, (2) through Hamilton (Hermitian) product and embedding propagation layers, we can obtain both latent inter- and intra- dependencies between components of users and items, (3) Hamilton (Hermitian) product with inner product can capture the intricate antisymmetric relation between users and items.

5.3 Hyper-parameter studies

5.3.1 Number of embedding propagation layer

In this section, we explore the influence of embedding propagation layer numbers on GCFC and GCFQ. We search layer numbers in {1, 2, 3, 4} for GCFC and GCFQ on four datasests. The experimental results are shown in Figures 2 and 3, and we can draw the following conclusions: (1) For the datasets Amazon-Cloth and Amazon-Music, the best number of layer is 1 for GCFC and GCFQ. This is because the Amazon-Cloth and Amazon-Music datasets are small, and one embedding propagation layer enables the node to obtain enough information from one-hop neighbors. (2) For the dataset Amazon-Electronic, the best number of layers is 3 for GCFC and 1 for GCFQ. Amazon-Electronic has more interaction information than Amazon-Cloth and Amazon-Music. Therefore, GCFC needs 3 layers to enhance the representation ability of model. Compared with GCFC, GCFQ has stronger modeling ability, so it only needs 1 layer for Amazon-Electronic. (3) The Book-Crossing is the largest in the four datasets and has more abundant interaction information. Therefore, GCFC and GCFQ need 3 and 4 embedding propagation layers respectively.

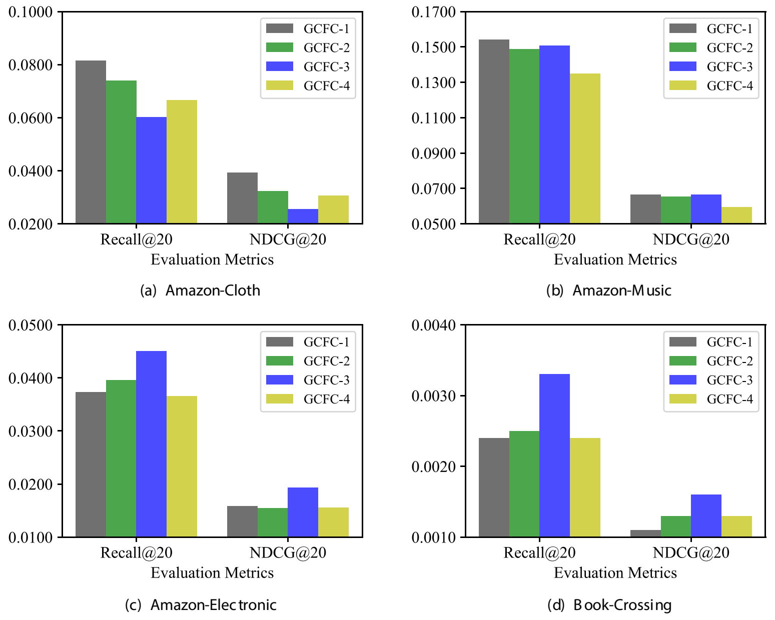

Figure 2 Effect of number of embedding propagation layer on GCFC

5.3.2 Effect of embedding dimensionality

In order to study the impact of embedding dimensionality on models, we search the embedding dimensionality in {16, 32, 64, 128, 256} for GCFC and GCFQ on four datasests. The

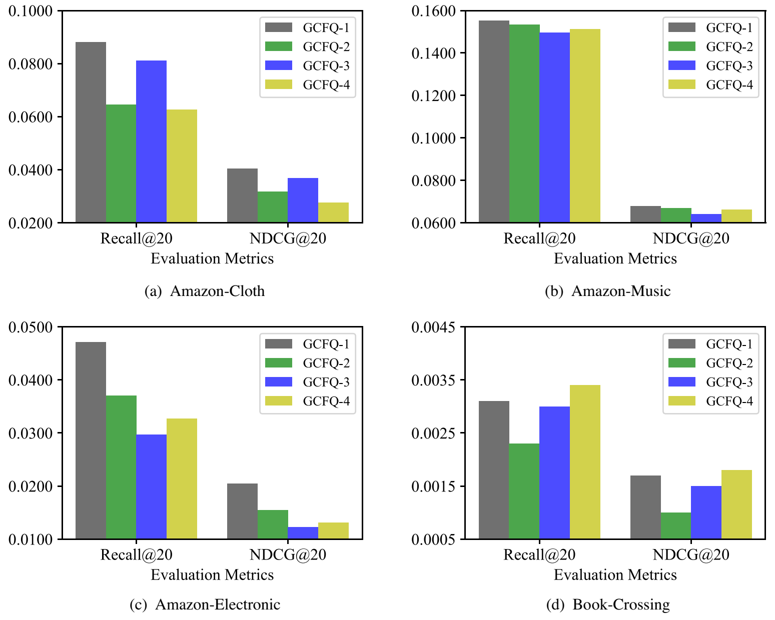

Figure 3 Effect of number of embedding propagation layer on GCFQ

experimental results are shown in Figures 4 and 5. From the figures, we can find that for Amazon-Cloth, Amazon-Music and Amazon-Electronic datasets, the best embedding dimensionality is 64. Higher dimensionality will lead to overfitting. For Book-Crossing, performance is improved as the dimensionality increase. This is because Book-Crossing has more interaction information than other three datasets, and requires a higher embedding dimensionality to represent users and items.

5.4 Model analysis

5.4.1 Space and time complexity analysis

For GCFC, it can be found from formulas (6) and (7) that we introduces embedding matrix $H_C^{(l)} \in \mathbb{C}^{(m+n)\times d_l}$ and trainable complex weight matrix $W_C^{(l)} \in \mathbb{C}^{(d_{l-1}/2)\times d_l}$ in layer l. We can find the space complexity of GCFC at each layer is $O(m+n+d_{l-1}/2)\times d_l$, where m and n are the number of users and items respectively; d_{l-1} and d_l are the embedding size at layer $l-1$ and l; d_{l-1} and d_l are smaller than m and n. Similarly, the space complexity of GCFQ at each layer is $O(m+n+d_{l-1}/4)\times d_l$. The proposed GCFC and GCFQ can be regarded as the expansion of NGCF in complex and quaternion spaces, so the space complexity of NGCF at each layer is about $O(m+n+d_{l-1})\times d_l$, which shows GCFC and GCFQ have similar space complexity with NGCF.

In terms of GCFC and GCFQ, the difference in time complexity mainly lies in nonlinear transformation. For GCFC, it can be found from formulas (6) and (7) that the matrix operation has computational complexity $O((m+n)\times(d_{l-1}+2)\times d_l)$ in nonlinear transformation for layer l. Similarly, the time complexity of GCFQ is $O((m+n)\times(d_{l-1}+4)\times d_l)$. The time complexity of NGCF is about $O((m+n)\times(d_{l-1}+1)\times d_l)$. For NGCF, GCFC and GCFQ, the time complexity difference is caused by inter-dependencies and intra-dependencies between components of users and items. From Section 5.2 we can see that in some cases GCFC

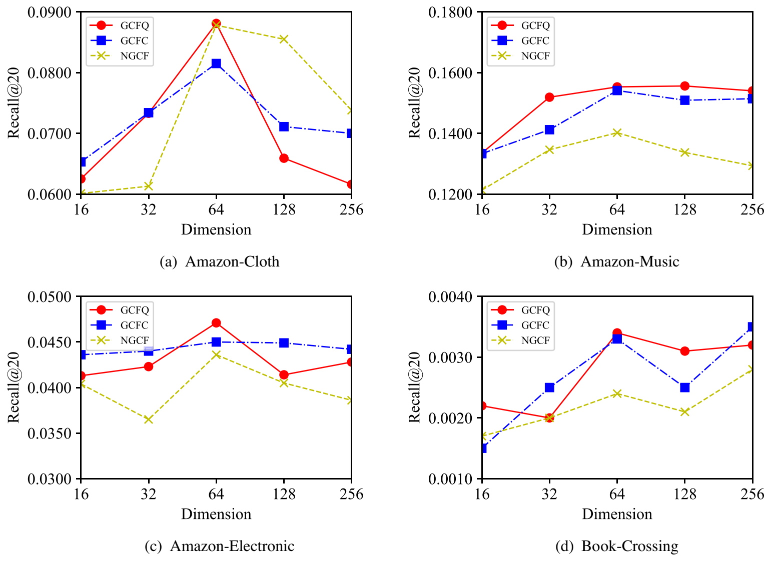

Figure 4 Effect of embedding dimensionality on GCFC and GCFQ with Recall@20

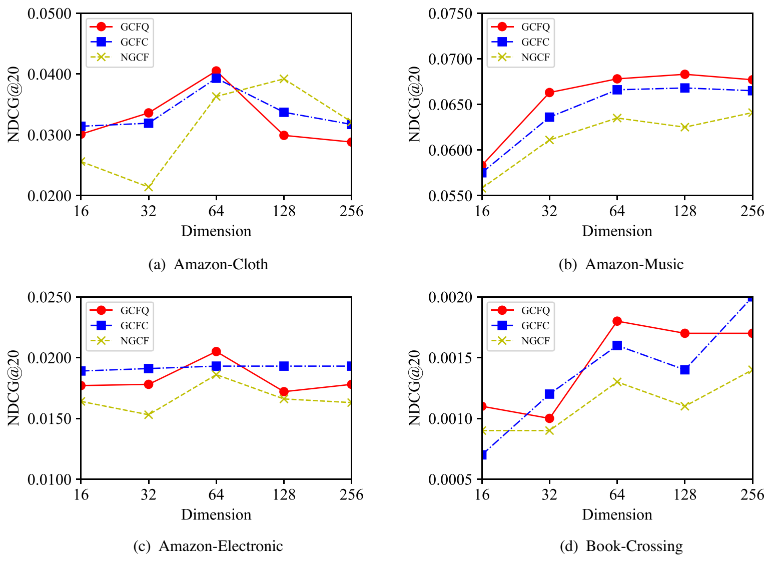

Figure 5 Effect of embedding dimensionality on GCFC and GCFQ with NDCG@20

achieves the best performance except for GCFQ. Therefore, if there is a requirement for the time of model, GCFC is a good choice compared with GCFQ.

5.4.2 Representational capacity of non-real embedding

The proposed GCFC and GCFQ can be regarded as the expansion of NGCF in complex and quaternion spaces. NGCF employs real representations as the embeddings of users and items. GCFC and GCFQ use complex representations and quaternion representations respectively as embeddings. In order to verify the representational capacity of non-real embedding, we compare the performance of GCFC and GCFQ with NGCF in different embedding dimension. From the Figures 4 and 5, we can find that: (1) For the dataset Amazon-Cloth, GCFQ and GCFC are better than NGCF before the best embedding dimension 64. As the dimension increases, the performance of three models decreases, and GCFQ and GCFC decrease faster. This is because the Amazon-Cloth dataset is smaller, and the overfitting of GCFQ and GCFC is more serious with the improvement of dimension. This also shows that the modeling ability of GCFQ and GCFC is stronger than NGCF. (2) For the datasets Amazon-Music, Amazon-Electronic and Book-Crossing, as embedding dimension increases, the performance of GCFQ and GCFC are always better than NGCF. This is because complex and quaternion embeddings have one real component and multiple imaginary components. Thanks to Hermitian and Hamilton products, we can obtain both latent inter- and intra- dependencies between components of embeddings. Therefore, compared with real embeddings, complex and quaternion embeddings have a better representational capacity.

5.4.3 Effect of Hermitian and Hamilton products in interaction modeling layer

Compared with the inner product, Hermitian and Hamilton products enable the components of embeddings of user and item to interact with each other, which endows them with better modeling capability while maintaining the simplicity. In addition, the Hermitian and Hamilton products are inherently asymmetrical, which enables them to capture the asymmetry in the recommendation. In order to verify the effectiveness of Hermitian and Hamilton products, we propose two variants of model: Model-in and Model-sys. Model-in replaces Hermitian or Hamilton product in interaction modeling layer with inner product. Model-sys redefines the prediction function in models. Specifically, when calculating the products between user and item in formulas (9) and (14), we redefine the products in GCFC-sys and GCFQ-sys as follows:

$$r_C = (\langle h_C^u, h_C^v \rangle + \langle h_C^v, h_C^u \rangle)/2 \tag{17}$$

$$r_Q = (h_Q^u \otimes h_Q^v + h_Q^v \otimes h_Q^u)/2 \tag{18}$$

Through above formula, we can eliminate the influence of asymmetry property of Hermitian product and Hamilton product on the model.

We conduct experiments on four datasets with metrics Recall@20 and NDCG@20, and the experimental results are shown in Table 4, from which we can draw the following conclusions: (1) Eliminating the Hermitian and Hamilton products in interaction modeling layer reduces the performance of model. This is because both latent inter-dependencies and intra-dependencies between the components of embeddings of user and item can be obtained through Hermitian and Hamilton products. The inner product only models the interaction of the corresponding components between user and item. Therefore, compared with Hermitian or Hamilton product, the inner product has a weaker modeling ability. (2) Eliminating the asymmetry of Hermitian and Hamilton products reduces the performance of model. In the recommendation system, users and items belong to different sets, and there is obvious asymmetry between them. If this asymmetry is not considered, the performance of the model will decrease. From Table 1, we can see that the asymmetry of the dataset Book-Crossing is very strong, and there are more than twice as many items as users. So, compared with the other two datasets, the performance of GCFC-sys and GCFQ-sys drop more in Book-Crossing.

6 Conclusion and future work

In this paper, we apply graph neural network to recommendation in complex and quaternion spaces, and propose two specific models GCFC and GCFQ respectively. These two models respectively employ complex and quaternion embeddings to represent users and items. Then based on Hermitian product (for GCFC) or Hamilton product (for GCFQ) and embedding

Table 4 Effect of Hermitian and Hamilton products on GCFC and GCFQ

	Amazon-Cloth		Amazon-Music		Amazon-Electronic		Book-Crossing	
Model	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
GCFC	**0.0815**	**0.0393**	**0.1541**	**0.0666**	**0.0450**	**0.0193**	**0.0033**	**0.0016**
GCFC-in	0.0807	0.0390	0.1535	0.0663	0.0435	0.0190	0.0029	0.0015
GCFC-sys	0.0802	0.0392	0.1514	0.0660	0.0439	0.0192	0.0030	0.0015
GCFQ	**0.0881**	**0.0405**	**0.1553**	**0.0678**	**0.0471**	**0.0205**	**0.0034**	**0.0018**
GCFQ-in	0.0820	0.0401	0.1551	0.0673	0.0413	0.0174	0.0029	0.0015
GCFQ-sys	0.0808	0.0346	0.1500	0.0663	0.0433	0.0186	0.0028	0.0014

propagation layers, we can obtain the intricate relations between users and items. Finally, in interaction modeling layer, we aggregate the embedding representations of different propagation layers and use Hermitian or Hamilton product with inner product to obtain the final prediction score between users and items. Extensive experimental results on four widely used datasets show the effectiveness of GCFC and GCFQ.

In the future work, we can further explore from many aspects. For example, for other recommendation scenarios, such as social recommendation, POI recommendation and interactive recommendation, we can model user and item on complex and quaternion spaces to further improve performance. In addition, many studies model recommendation scenarios as heterogeneous information networks. How to apply complex and quaternion space to heterogeneous information networks and reduce the time of model is a worthy research direction. Furthermore, in addition to complex and quaternion spaces, other non-real spaces, such as octonion space, are also worthy of exploration on how to apply them in recommendation scenarios. Finally, when we take text and image about user and item into account, how to model multimodal information and user item interaction in complex and quaternion spaces is a challenging direction.

Acknowledgements The work was supported by National Natural Science Foundation of China (62172086, 62272092, 62106039).

Author Contributions Longcan Wu: Conceptualization, Methodology, Software, Writing Original draft preparation. Daling Wang: Supervision, Validation, Funding acquisition. Shi Feng: Supervision, Validation, Funding acquisition. Xiangmin Zhou: Supervision, Validation, Funding acquisition. Yifei Zhang: Supervision, Validation, Funding acquisition. Ge Yu: Supervision, Project administration, Funding acquisition.

Funding The work was supported by National Natural Science Foundation of China (62172086, 62272092, 62106039).

Availability of data and materials Please refer to Section 5.1.1 of the paper.

Declarations

Ethical Approval This declaration is "not applicable".

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173-182 (2017)
2. Xue, H., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 3203-3209 (2017)
3. Liao, J., Zhou, W., Luo, F., Wen, J., Gao, M., Li, X., Zeng, J.: Sociallgn: Light graph convolution network for social recommendation. Inf. Sci. **589**, 595–607 (2022)
4. Yu, B., Li, X., Fang, J., Tai, C., Cheng, W., Xu, J.: Memory-augmented meta-learning framework for session-based target behavior recommendation. World Wide Web (WWW) **26**(1), 233–251 (2023)
5. Li, Y., Guo, X., Lin, W., Zhong, M., Li, Q., Liu, Z., Zhong, W., Zhu, Z.: Learning dynamic user interest sequence in knowledge graphs for click-through rate prediction. IEEE Trans. Knowl. Data Eng. **35**(1), 647–657 (2023)
6. van den Berg, R., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)

7. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM International Conference on Knowledge Discovery & Data Mining, pp. 974-983 (2018)
8. Parcollet, T., Morchid, M., Linarès, G.: A survey of quaternion neural networks. Artif. Intell. Rev. **53**(4), 2957–2982 (2020)
9. Lee, C., Hasegawa, H., Gao, S.: Complex-valued neural networks: A comprehensive survey. IEEE CAA J. Autom. Sin. **9**(8), 1406–1426 (2022)
10. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 101-109 (2019)
11. Rendle, S., Krichene, W., Zhang, L., Anderson, J.R.: Neural collaborative filtering vs. matrix factorization revisited. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 240-248 (2020)
12. Sun, Z., Yu, D., Fang, H., Yang, J., Qu, X., Zhang, J., Geng, C.: Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 23-32 (2020)
13. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning, pp. 2071-2080 (2016)
14. Zhang, S., Yao, L., Tran, L.V., Zhang, A., Tay, Y.: Quaternion collaborative filtering for recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 4313-4319 (2019)
15. Gao, M., Chen, L., He, X., Zhou, A.: Bine: Bipartite network embedding. In: The 41st International ACM Conference on Research & Development in Information Retrieval, SIGIR, pp. 715-724 (2018)
16. Huang, W., Li, Y., Fang, Y., Fan, J., Yang, H.: Biane: Bipartite attributed network embedding. In: Proceedings of the 43rd International ACM Conference on Research and Development in Information Retrieval, SIGIR, pp. 149-158 (2020)
17. Wu, L., Wang, D., Feng, S., Zhou, X., Zhang, Y., Yu, G.: Graph collaborative filtering for recommendation in complex and quaternion spaces. In: Web Information Systems Engineering, vol. 13724, pp. 579-594 (2022)
18. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
19. Yang, J., Chen, C., Wang, C., Tsai, M.: Hop-rec: high-order proximity for implicit recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 140-144 (2018)
20. Chen, C., Wang, C., Tsai, M., Yang, Y.: Collaborative similarity embedding for recommender systems. In: Proceedings of the 28th International Conference on World Wide Web, pp. 2637-2643 (2019)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations (2017)
22. Zhu, T., Sun, L., Chen, G.: Graph-based embedding smoothing for sequential recommendation. IEEE Trans. Knowl. Data Eng. **35**(1), 496–508 (2023)
23. Zhai, P., Yang, Y., Zhang, C.: Causality-based CTR prediction using graph neural networks. Inf. Process. Manag. **60**(1), 103137 (2023)
24. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM Conference on Research and Development in Information Retrieval, pp. 165-174 (2019)
25. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: A survey. ACM Comput. Surv. **55**(5), 97–19737 (2023)
26. Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y., Pal, C.J.: Deep complex networks. In: Proceedings of the 6th International Conference on Learning Representations (2018)
27. Yang, M., Ma, M.Q., Li, D., Tsai, Y.H., Salakhutdinov, R.: Complex transformer: A framework for modeling complex-valued sequence. In: Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4232-4236 (2020)
28. Wisdom, S., Powers, T., Hershey, J.R., Roux, J.L., Atlas, L.E.: Full-capacity unitary recurrent neural networks. In: Annual Conference on Neural Information Processing Systems, pp. 4880-4888 (2016)
29. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: Proceedings of the 7th International Conference on Learning Representations (2019)
30. Parcollet, T., Morchid, M., Linarès, G.: Quaternion convolutional neural networks for heterogeneous image processing. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 8514-8518 (2019)

31. Shi, L., Geng, S., Shuang, K., Hori, C., Liu, S., Gao, P., Su, S.: Multi-layer content interaction through quaternion product for visual question answering. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, pp. 4412-4416 (2020)
32. Nguyen, D.Q., Nguyen, T.D., Phung, D.Q.: Quaternion graph neural networks. In: Asian Conference on Machine Learning, vol. 157, pp. 236-251 (2021)
33. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems, pp. 2731-2741 (2019)
34. Tay, Y., Zhang, A., Luu, A.T., Rao, J., Zhang, S., Wang, S., Fu, J., Hui, S.C.: Lightweight and efficient neural natural language processing with quaternion networks. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, pp. 1494-1503 (2019)
35. Fang, Y., Zhao, P., Xian, X., Fang, J., Liu, G., Liu, Y., Sheng, V.S.: Quaternion-based graph contrastive learning for recommendation. In: International Joint Conference on Neural Networks, pp. 1-8 (2022)
36. Tran, T., You, D., Lee, K.: Quaternion-based self-attentive long short-term user preference encoding for recommendation. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 1455-1464 (2020)
37. Li, Z., Xu, Q., Jiang, Y., Cao, X., Huang, Q.: Quaternion-based knowledge graph network for recommendation. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 880-888 (2020)
38. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 639-648 (2020)
39. Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M.: Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 27-34 (2020)
40. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (2015)