

## Article

# Multi-Level Knowledge-Aware Contrastive Learning Network for Personalized Recipe Recommendation

Zijian Bai <sup>1</sup>, Yinfeng Huang <sup>2</sup>, Suzhi Zhang <sup>1,\*</sup>, Pu Li <sup>1,\*</sup>, Yuanyuan Chang <sup>3</sup> and Xiang Lin <sup>1</sup>

<sup>1</sup> School of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China

<sup>2</sup> School of Engineering, University of Glasgow, Glasgow G128QQ, UK

<sup>3</sup> School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China

\* Correspondence: zhsuzhi@zzuli.edu.cn (S.Z.); lipu@zzuli.edu.cn (P.L.)

**Abstract:** Personalized recipe recommendation is attracting more and more attention, which can help people make choices from the exploding growth of online food information. Unlike other recommendation tasks, the target of recipe recommendation is a non-atomic item, so attribute information is especially important for the representation of recipes. However, traditional collaborative filtering or content-based recipe recommendation methods tend to focus more on user–recipe interaction information and ignore higher-order semantic and structural information. Recently, graph neural networks (GNNs)-based recommendation methods provided new ideas for recipe recommendation, but there was a problem of sparsity of supervised signals caused by the long-tailed distribution of heterogeneous graph entities. How to construct high-quality representations of users and recipes becomes a new challenge for personalized recipe recommendation. In this paper, we propose a new method, a multi-level knowledge-aware contrastive learning network (MKCLN) for personalized recipe recommendation. Compared with traditional comparative learning, we design a multi-level view to satisfy the requirement of fine-grained representation of users and recipes, and use multiple knowledge-aware aggregation methods for node fusion to finally make recommendations. Specifically, the local-level includes two views, interaction view and semantic view, which mine collaborative information and semantic information for high-quality representation of nodes. The global-level learns node embedding by capturing higher-order structural information and semantic information through a network structure view. Then, a kind of self-supervised cross-view contrastive learning is invoked to make the information of multiple views collaboratively supervise each other to learn fine-grained node embeddings. Finally, the recipes that satisfy personalized preferences are recommended to users by joint training and model prediction functions. In this study, we conduct experiments on two real recipe datasets, and the experimental results demonstrate the effectiveness and advancement of MKCLN.

**Keywords:** personalized recipe recommendation; contrastive learning network; high-quality representation; multi-level view



**Citation:** Bai, Z.; Huang, Y.; Zhang, S.; Li, P.; Chang, Y.; Lin, X. Multi-Level Knowledge-Aware Contrastive Learning Network for Personalized Recipe Recommendation. *Appl. Sci.* **2022**, *12*, 12863. <https://doi.org/10.3390/app122412863>

Academic Editors: Shoujin Wang and Qi Zhang

Received: 4 November 2022

Accepted: 12 December 2022

Published: 14 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

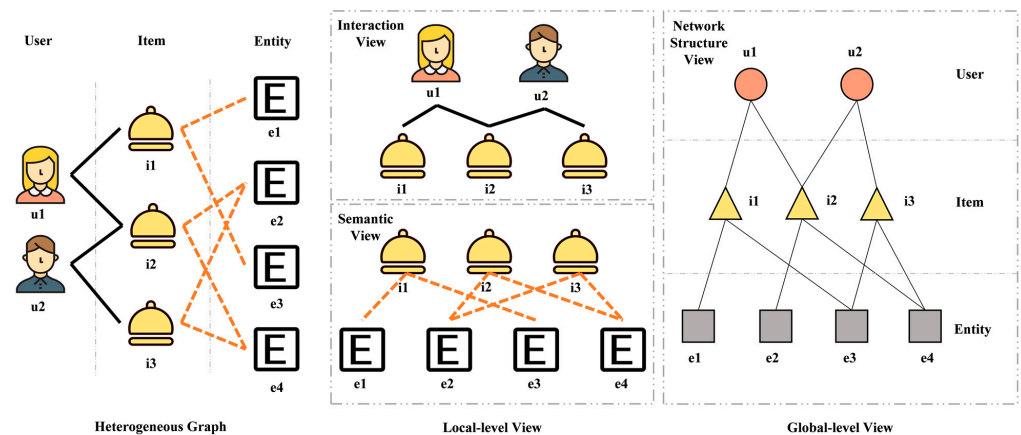
The growth of massive amounts of data make it difficult for users to select interactive items that meet their requirements, and this is no exception when it comes to recipe selection. Recipe websites (e.g., Allrecipes, Food.com (accessed on 4 November 2022), etc.) provide users with a rich selection of recipe content, reviews, categories, etc. However, with the huge number of users and the entries matching problem, it is no doubt a difficult challenge to recommend recipes that match users' personalized preferences from the large number of matching recipes. Therefore, personalized recipe recommendation aims to help users select recipes that meet their personal requirements from the information explosion of items.

Indeed, existing research on recommender systems can be divided into three categories: collaborative filtering-based methods (CF) [1,2], content-based methods (CB) [3,4] and hybrid methods [5,6]. Collaborative filtering-based recommendation methods learn representations of users and items from users' historical interactions, which are often easy to implement but lead to coarse-grained representations. Content-based recommendation methods construct representations of items by exploring their content information (e.g., photos, attributes, etc.); these methods ignore users' interests and cannot satisfy personalization requirements. Hybrid recommendation methods achieved good results in many application scenarios by combining models to address the shortcomings of different recommendation methods. There also exists much research on the task of recipe recommendation that experiments with three recommendation methods [4,5]. However, there are challenges in recipe recommendation that differ from the general recommendation task: (1) the target of recipe recommendation is non-atomic items, and the ingredients, flavors, cooking methods and other attributes of recipes are crucial to users' choices. This means that the recipe recommendation task should pay more attention to high-order relationship structure information and semantic information between nodes. (2) The personalized preferences of users in recipe recommendation are more complex. Two different users may have different reasons for choosing the same recipe, and there are differences in the taste of two recipes with similar ingredients. So, the personalized recipe recommendation task has higher quality requirements for fine-grained embedding representation of users and recipe nodes.

Graph neural networks (GNNs) were successfully applied to various fields in recommendation, such as: travel [7], video [3], etc., and received a lot of attention from researchers. Graph structure data can better represent the rich relationships between nodes, and at the same time, the powerful information aggregation capability of GNNs can enable the transfer of some features of neighboring nodes to the target node for a higher quality node representation. GNN-based models sometimes achieve an increase in the accuracy of recommendations at the cost of amplifying biases in the data and producing unfair recommendations [8], which brings new ideas for recipe recommendations. However, there are many difficulties in applying GNN to recipe recommendation as well. Among them, the long-tailed distribution of heterogeneous graph entities may cause the loss of the personalized value of recommendation results because the sparsity of supervised features may lead to the over-smoothing of node embeddings. For this reason, the recipe recommendation task needs to make full use of the limited interaction and relevance information for personalized recommendations.

In this paper, we propose a new multi-level knowledge-aware contrastive learning network (MKCLN) for personalized recipe recommendation. For the task of the personalized recommendation of recipes, we emphasize the exploration of multiple-level views that are explored to make full use of higher-order structural information in heterogeneous graphs and local collaborative and semantic information to learn higher quality node embedding representations. Specifically, first we regard the full graph as multiple views in two levels, global-level and local-level, as shown in Figure 1. Compared with the traditional contrast learning corruption or dropping to generate a contrast view, this complementary view approach preserves the complete structural information to the maximum extent and makes the learning node representation more fine-grained. In the local-level view, to better capture the local collaborative and semantic information, we use different aggregations to learn the embedding representation of user nodes and recipe nodes containing local structural information in the user–recipe graph and recipe–property graph. In the global-level view, we use a path-aware approach to learn node embedding by passing messages along multiple paths from the network structure view for the purpose of capturing higher-order structural information. Then, we introduce self-supervised cross-view contrastive learning so that the information of multiple views collaboratively supervise each other and learn nodes with high quality final embedding. Finally, the recipes satisfying personalized preferences are matched with users through joint training and model prediction functions to complete the recommendation task. The main contribution of this paper can be as follows:

- We emphasize the importance of high-order relational information and complex personalized requirements in recipe recommendation tasks, and introduce the idea of graph comparison learning for recipe recommendation. To our knowledge it is the first attempt on this problem.
- We propose multi-level knowledge-aware contrastive learning network (MKCLN), a heterogeneous graph learning model for personalized recipe recommendation. The model leverages the collaborative, semantic, and higher-order structural information of heterogeneous graphs by constructing multiple views at both global and local levels. Self-supervised cross-view contrastive learning is applied to multiple views to obtain comprehensive high-quality node embeddings and personalized recipe recommendations.
- We conducted extensive experiments on two different food datasets containing different national preferences, and the results demonstrate the effectiveness and sophistication of MKCLN.



**Figure 1.** A sample of the MKCLN model with two levels and multiple views.

## 2. Related Work

We review recipe recommendation, graph neural networks-based recommendation, and contrastive learning related research in this section.

### 2.1. Recipe Recommendation

On recipe analysis websites containing large amounts of data, recommendations are the most effective technique to satisfy users' personalized requirements for selection. Examples of recipe recommendation include collaborative filtering-based methods. Pecunne et al. [9] applied the traditional collaborative filtering methods BPR and LMF to the recipe recommendation domain with good results. Trattner et al. [4] applied the popular collaborative filtering in the LibRec framework to recipe recommendation, where the LDA method was improved in mining the personalized preference requirements of users. Collaborative filtering-based methods are usually easy to implement, but they ignore the rich content representation of recipes, such as ingredients, flavors, etc. However, they do not allow users to personalize their preferences. Content-based recipe recommendations focus on the content representation of recipes. Vivek et al. [10] and Khan et al. [11] focused on the similarity of recipes with the same rating for different users and recommended recipes as content information to different people. Gao et al. [12] proposed a hierarchical attention mechanism to embed images, ingredients, etc. into recipe representations to get more diverse recipe embeddings. These content-based methods improve in precision but neglect to make internal correlations between recipes and ingredients. Recently, several researches attempted to use graph-based methods for recipe recommendation. Gao et al. [5] used a graph convolutional network to model recipe data and obtained comprehensive embedding representations of recipes by the aggregation of three subgraphs. Tian et al. [13] proposed a graph learning approach to capture recipe content and collaboration signals through a heterogeneous graph neural network with hierarchical attention and a compo-

ment set transformer with good results. However, this method places more emphasis on collaborative information and ignores the value of higher-order structural information for recipes. Graph-based methods possess a great improvement in recipe recommendation accuracy, which indicates the advantage of structured graph data for recipe recommendation. However, due to the specificity of heterogeneous graphs, weighing local feature information with global high-order structural information becomes one of the challenges for recommendation.

## 2.2. Graph Neural Networks-Based Recommendation

Recently, graph neural networks were successfully applied to a large number of scenarios as a popular method for processing graph data. GNN-based recommendation uses the efficient information aggregation mechanism of GNN to aggregate features of multi-hop neighbors in the target node to capture the higher-order potential features of the node. Wang et al. [14] proposed KGCN, which applies graph convolutional neural networks to knowledge graphs by stacking multiple convolutional layers so that the neighboring feature information is continuously aggregated to the target nodes to obtain node embeddings. Ma et al. [15] proposed two meta-path-based proximity measures to jointly update node embeddings in heterogeneous graphs, achieving good recommendation results. He et al. [16] proposed a simplified and improved LightGCN that employed the neighbor aggregation component of the GCN in collaborative filtering and achieved better results. Wang et al. [17] developed the knowledge graph attention network (KGAT), which introduces attention mechanisms into the graph structure, connecting users and items by their attributes, and extracting higher-order linking paths to represent nodes in the network. However, the existing GNN-based recommendation mainly focuses on modeling the interaction information and ignores the collaboration information between entities.

## 2.3. Contrastive Learning

Contrastive learning attracted wide attention for its successful applications in computer vision [18], natural language processing [19], and video analysis [20] by comparing positive and negative samples from different perspectives on learning node representations. Wu et al. [21] performed embedding learning of nodes on heterogeneous graphs by discarding nodes with edges or random wandering strategies, but the comparative learning method by impairing the graph structure is not suitable for recommendations, and it is difficult for this method to capture higher-order potential preference information to the extent that the recommendations are depersonalized. Wang et al. [22] proposed HeCo, a cross-view self-supervised contrast learning network, which is able to portray the local and higher-order structures of the network to obtain high-quality embedding representations. Therefore, it has great potential to attempt to apply the advantages of self-supervised learning to the recipe recommendation task using a cross-view contrast learning method.

## 3. Problem Formulation

The main task objective of this paper is to recommend recipes to users that match their personalized preferences. Given a set of users  $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$  and recipes  $\mathcal{R} = \{r_1, r_2, \dots, r_N\}$ , the interaction matrix is denoted as  $\mathbf{Y} \in \mathbb{R}^{M \times N}$ , where  $M, N$  denote the number of users and recipes, respectively. The user–recipe interaction diagram can be represented as  $\mathcal{G}_{ui} = \{(u, y_{ur}, r) \mid u \in \mathcal{U}, r \in \mathcal{R}\}$ , where  $y_{ur}$  denotes whether user  $u$  performs an interaction behavior (e.g., commenting or making) with recipe  $r$ , as defined by:

$$y_{ur} = \begin{cases} 1, & \text{if user } u \text{ interacted with recipe } r \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that some research shows that sentiment analysis of user review information and incorporation into the recommendation system is beneficial to improve the recommendation quality [23]. In some other studies, instead of using user interactions (binary values), multi-

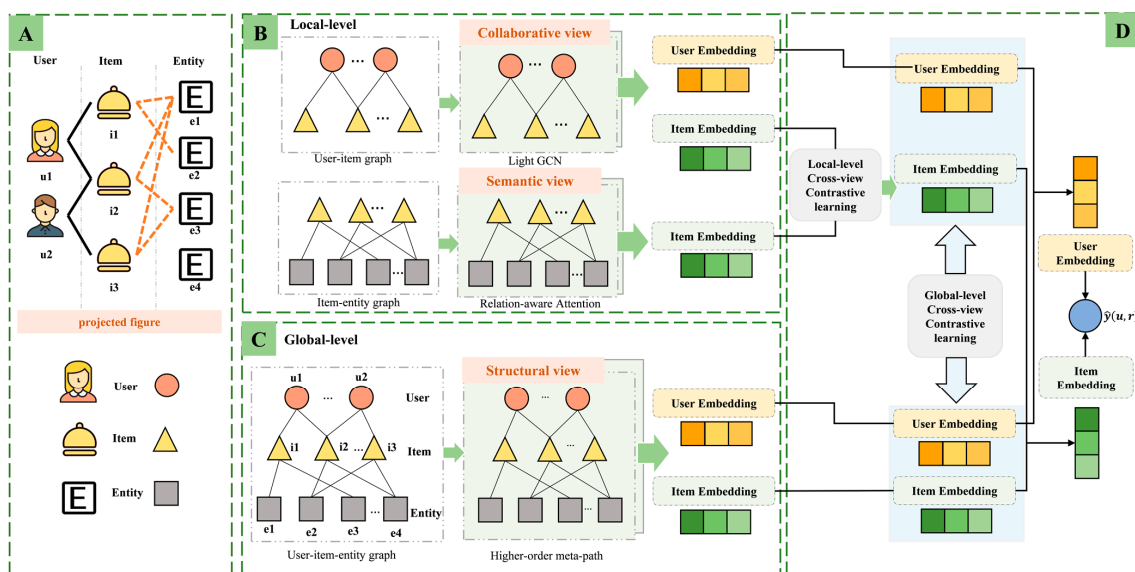
valued ratings are used. These ratings can either be obtained explicitly from users or can be obtained by applying sentiment analysis techniques on comments or reviews on the items. However, due to the low quality of explicit feedback on the recipe dataset, we assign it different weights and treat it as interaction in the preprocessing process. We define the knowledge-aware heterogeneous graph  $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$  to denote a real-world recipe containing rich attributes, where the entity–relation–entity triplet  $(h, r, t)$  denotes the head entity  $h$  connected to the tail entity  $t$  by the relation  $r$ .  $\mathcal{E}$  and  $\mathcal{R}$  denote the set of entities and relationships. In the knowledge-aware recipe heterogeneous graph, the recipe contains rich attributes and relationships such as ingredients, tastes, cooking methods, and types. The task goal of recipe recommendation is to recommend recipes to users by the learned function  $\hat{y} = \mathcal{F}(\mathcal{G}, \Theta)$ , where  $\Theta$  is a model parameter. We specify the task as follows:

Input: Users  $\mathcal{U}$ , recipe  $R$ , user–recipe interactions  $\mathcal{Y}$ , heterogeneous graph  $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ .

Output: A predictor function  $\hat{y} = \mathcal{F}(\mathcal{G}, \Theta)$ , which estimates the probability of user  $u$  interaction with the recipe  $r$ .

#### 4. Methodology

In this section, we present in detail the multi-level knowledge-aware contrastive learning network for personalized recipe recommendation model, as shown in Figure 2. The figure shows the three important parts of MKCLN: (1) Heterogeneous graph multi-level view encoder network. This component generates several different graph views from the full graph at local-level and global-level, which includes interaction view, semantic view, and network structure view. Additionally, it captures node collaboration information, semantic information, and higher-order structure information to obtain a fine-grained embedding representation of users and recipes. (2) Graph cross-view contrastive optimization. This part first performs cross-view contrastive learning between two local views, and then learns the result with the global view across views to obtain high-quality global embedding representations considering the information of each part. (3) Model prediction and joint training. We recommend the recipes that users might interact with by prediction function, and optimize the model using the joint training method.



**Figure 2.** A sample of the MKCLN model with two levels and multiple views. Our proposed overall architecture of MKCLN. (A) Denotes the heterogeneous graph, (B,C) denote the two levels in the heterogeneous graph multi-level view encoder network, and (D) denotes the graph cross-view contrastive optimization and prediction function.



#### 4.1. Heterogenous Graph Multi-Level View Encoder Network

Due to the specificity of the recipe recommendation task, traditional comparison learning corruption or dropping to generate comparison views is not applicable to recipe recommendations. Because unaccounted factors may be the reason for determining the user's choice, coarse-grained representations may lead to biased recommendation results. Therefore, we propose a heterogenous graph multi-level view encoder network for embedding different nodes in recipe recommendation. First, we consider capturing collaborative and semantic information of nodes through local interaction view and semantic view, here we are more focused on the low-order relationship between the user–recipe graph and recipe–entity graph that depends on edge connection. After that, we explore the higher-order structure information in the graph by global-level network structure view, where we pay more attention to the long-range connectivity and multi-hop neighbor feature information in the graph. Finally, we generate embeddings of different views of the nodes as input for comparison learning.

##### 4.1.1. Local-Level View Encoder

Local-level mainly captures the collaborative information and semantic information of nodes through two complementary views. We use different encoding methods to embed the interaction view and semantic view individually to ensure that the embedded representation contains the complete content.

**Interaction view Encoder.** Interaction view is mainly designed to capture collaborative information in the user behavior record graph, using the historical interaction connections between users and recipes as the basis for modeling. For the interaction view, the relationship of interaction behaviors is relatively single, and we consider an effective and efficient aggregation method for message aggregation and delivery. We choose the message propagation strategy of Light-GCN [16] for encoding, and its effective lightweight architecture can be well applied with user–recipe collaborative information collection. The specific form is as follows:

$$\mathbf{x}_u^{(l+1)} = \sum_{r \in N_u} \frac{\mathbf{x}_r^{(l)}}{\sqrt{|N_u||N_r|}}; \mathbf{x}_r^{(l+1)} = \sum_{u \in N_r} \frac{\mathbf{x}_u^{(l)}}{\sqrt{|N_r||N_u|}} \quad (2)$$

where  $\mathbf{x}_u^l$  and  $\mathbf{x}_r^l$  are the embedding representations of user  $u$  and recipe  $r$  at layer  $l$ .  $N_u$  and  $N_r$  denote the set of recipes  $r$  that interact with user  $u$  and the set of users connected to recipe  $r$ , respectively. After that, we sum the embedded expressions for different layers as the output of the interaction view, as follows:

$$\mathbf{e}_u^i = \mathbf{x}_u^{(0)} + \dots + \mathbf{x}_u^{(l)}; \mathbf{e}_r^i = \mathbf{x}_r^{(0)} + \dots + \mathbf{x}_r^{(l)} \quad (3)$$

where  $\mathbf{e}_u^i$  and  $\mathbf{e}_r^i$  are the node embedding containing the collaboration information after encoding by the interaction view.

**Semantic View Encoder.** Semantic view is mainly designed to capture the semantic information between recipes and properties. The properties corresponding to recipes in recipe recommendations often contain nodes with multiple relationship types, such as: ingredients, tastes, cooking methods, etc. Inspired by the graph attention mechanism [7,17], we use a relation-aware embedding layer to encode the semantic view as follows:

$$\mathbf{x}_r^s = \sigma \left( \sum_{e \in N_r^e} \alpha_{e,r} \cdot \mathbf{x}_e \right) \quad (4)$$

$$\alpha_{e,r} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_\gamma \cdot [\mathbf{x}_e \parallel \mathbf{x}_r]))}{\sum_{e \in N_r^e} \exp(\text{LeakyReLU}(\mathbf{a}_\gamma \cdot [\mathbf{x}_e \parallel \mathbf{x}_r]))} \quad (5)$$

where  $\sigma$  is the nonlinear activation function,  $x_e$  is the entity embedding, and  $\alpha_{e,r}$  denotes the attention relevance of entities and relationships under different relationships. Additionally,  $\mathbf{a}_r$  denotes the relation-aware attention vector, and  $N_r^e$  is the entity neighbor of different relationships connected to the recipe  $r$ . In this manner we encode the semantic information that the recipe nodes contain multiple relationship entity types, which is important for us to determine the user's preferences for the recipes. After that we define the importance score  $w_{r,a}$  of the relationship to the node and normalize it as follows:

$$\begin{aligned} w_{r,a} &= \frac{1}{|V_a|} \sum_{r \in V_a} q^\top \cdot \tanh(\mathbf{W}_s x_r^s + \mathbf{b}_s) \\ \beta_{r,a} &= \frac{\exp(w_{r,a})}{\sum_{r=1}^S \exp(w_{r,a})} \end{aligned} \quad (6)$$

where  $V_a$  denotes the set of nodes associated with relation  $a$  and  $q$  denotes the relational level attention vector. The matrix  $\mathbf{W}_s$  and the bias  $\mathbf{b}_s$  are parameters that can be learned.  $\beta_{r,a}$  can be regarded as the degree of contribution of the relation  $a$  to the recipe nodes, and  $S$  denotes the number of recipe vectors. Ultimately, we aggregate the recipe embedding of different relationships in the semantic view as follows:

$$e_r^s = \sum_{m=1}^S \beta_{r,a} \cdot x_r^s \quad (7)$$

#### 4.1.2. Global-Level View Encoder

We obtained embeddings of users and recipe nodes from the local-level view, but ignored the higher-order feature information in the graph. The purpose of the global-level view is to capture higher-order structural information by exploring long-range connections in heterogeneous graphs that may contain potential preferred content for users.

**Global-level network structure view encoder.** We introduce a higher-order meta-path method for full graph node encoding, which contains both higher-order structural and global semantic information. Specifically, the set of meta-paths that exist from node  $i$  is defined as  $P = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m\}$ , where  $\mathcal{P}_n \in P$ . For different types of node  $i$  meta-paths are set differently, for example, for recipe nodes there are two meta-paths: recipe–entity–recipe and recipe–user–recipe meta-paths. We use a GCN [22,24] to encode the features in the following form:

$$x_i^{\mathcal{P}_n} = \frac{1}{d_i + 1} x_i + \sum_{j \in N_i^{\mathcal{P}_n}} \frac{1}{\sqrt{(d_i + 1)(d_j + 1)}} x_j \quad (8)$$

where  $d_i$  and  $d_j$  denote the degrees of node  $i$  and node  $j$ , respectively, and  $N_i^{\mathcal{P}_n}$  denotes the set of neighbors of node  $i$  based on meta-paths. After going through  $m$  meta-paths, we fuse the embeddings of  $m$  nodes  $i$  using an attention mechanism to ensure the different semantic effects on the node expressions, formally as follows:

$$e_i^g = \sum_{n=1}^m \beta_{\mathcal{P}_n} \cdot x_i^{\mathcal{P}_n} \quad (9)$$

where  $\beta_{\mathcal{P}_n}$  denotes the weights of different meta-paths and is calculated as follows:

$$\begin{aligned} w_{\mathcal{P}_n} &= \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_g^\top \cdot \tanh(\mathbf{W}_g x_i^{\mathcal{P}_n} + \mathbf{b}_g), \\ \beta_{\mathcal{P}_n} &= \frac{\exp(w_{\mathcal{P}_n})}{\sum_{i=1}^m \exp(w_{\mathcal{P}_i})}, \end{aligned} \quad (10)$$

where  $V$  is the set of target nodes and  $\mathbf{a}_g$  denotes the global-level attention vector. The matrix  $\mathbf{W}_g$  and the bias  $\mathbf{b}_g$  are the parameters that can be learned. At this point, we obtain the node embedding of global-level as follows:

$$e_i^g = \begin{cases} e_u^g, & \text{if node } i \text{ is the user node} \\ e_r^g, & \text{if node } i \text{ is the recipe node} \end{cases} \quad (11)$$

#### 4.2. Graph Cross-View Contrastive Optimization

With the heterogeneous graph multi-level view encoder network we obtain the embedding representation from three views for different perspectives of the nodes, and then we obtain the final fine-grained embedding representation by cross-view contrastive learning.

**Local-level Contrast Learning.** In the local-level view, the main purpose is to contrast learning of the two embedding expressions  $e_r^i$  and  $e_r^s$  of the recipe. First, we feed the embeddings of the two views into an MLP with hidden layers so that they are mapped into a uniform space as follows:

$$\begin{aligned} z_{r-p}^i &= W^{(2)} \sigma \left( W^{(1)} e_r^i + b^{(1)} \right) + b^{(2)} \\ z_{r-p}^s &= W^{(2)} \sigma \left( W^{(1)} e_r^s + b^{(1)} \right) + b^{(2)} \end{aligned} \quad (12)$$

where  $\{W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}\}$  are the parameters that can be trained and they are shared by two views.  $\sigma$  is the ELU non-linear function. After that we need to define positive and negative samples for learning, inspired by other applications of comparative learning [25,26], we define positive and negative samples as shown in Figure 3. The same node of another view for the target node is treated as the positive sample, the other nodes of the same view are treated as the intra-view negative sample, and the nodes of another view except for the positive sample are treated as the inter-view negative sample. From the defined positive and negative samples, we have contrastive loss as follows:

$$\mathcal{L}_{local} = -\log \frac{\exp \left( \frac{\text{sim}(z_{r-p}^s, z_{r-p}^i)}{\tau} \right)}{\exp \left( \frac{\text{sim}(z_{r-p}^s, z_{r-p}^i)}{\tau} \right) + \sum_{k \neq i} \exp \left( \frac{\text{sim}(z_{r-p}^s, z_{k-p}^s)}{\tau} \right) + \sum_{k \neq i} \exp \left( \frac{\text{sim}(z_{r-p}^s, z_{k-p}^i)}{\tau} \right)} \quad (13)$$

where  $\text{sim}(\cdot, \cdot)$  denotes the cosine similarity calculation of the two vectors and  $\tau$  denotes the temperature parameter. In Equation (13),  $\exp(\text{sim}(z_{r-p}^s, z_{r-p}^i)/\tau)$  denotes the positive sample pair,  $\sum_{k \neq i} \exp(\text{sim}(z_{r-p}^s, z_{k-p}^s)/\tau)$  denotes the intra-view negative sample pair, and  $\sum_{k \neq i} \exp(\text{sim}(z_{r-p}^s, z_{k-p}^i)/\tau)$  denotes the inter-view negative sample pair. We obtained the local fine-grained embedding representation by local-level cross-view comparison learning.

**Global-level Contrast Learning.** After local contrast learning, we obtain the final node representation by performing cross-view contrastive learning of the local-level view embedding with the global-level view embedding. First mapping them to the same space, we still do it by an MLP with a hidden layer:

$$\begin{aligned} z_{r-p}^g &= W^{(2)} \sigma \left( W^{(1)} e_r^g + b^{(1)} \right) + b^{(2)} \\ z_{r-p}^l &= W^{(2)} \sigma \left( W^{(1)} (e_r^i + e_r^s) + b^{(1)} \right) + b^{(2)} \end{aligned}$$

Similarly, we use cross-view positive and negative samples for contrastive learning and define the contrastive loss as follows:



$$\mathcal{L}_r^g = -\log \frac{\exp\left(\frac{\text{sim}(z_{r-p}^g, z_{r-p}^l)}{\tau}\right)}{\exp\left(\frac{\text{sim}(z_{r-p}^g, z_{r-p}^l)}{\tau}\right) + \sum_{k \neq i} \exp\left(\frac{\text{sim}(z_{r-p}^g, z_{k-p}^g)}{\tau}\right) + \sum_{k \neq i} \exp\left(\frac{\text{sim}(z_{r-p}^g, z_{k-p}^l)}{\tau}\right)} \quad (14)$$

$$\mathcal{L}_r^l = -\log \frac{\exp\left(\frac{\text{sim}(z_{r-p}^l, z_{r-p}^g)}{\tau}\right)}{\exp\left(\frac{\text{sim}(z_{r-p}^l, z_{r-p}^g)}{\tau}\right) + \sum_{k \neq i} \exp\left(\frac{\text{sim}(z_{r-p}^l, z_{k-p}^l)}{\tau}\right) + \sum_{k \neq i} \exp\left(\frac{\text{sim}(z_{r-p}^l, z_{k-p}^g)}{\tau}\right)} \quad (15)$$

where  $\mathcal{L}_r^g$  denotes the contrastive learning loss of recipe nodes for the global-level view, and  $\mathcal{L}_r^l$  denotes the contrastive learning loss of recipe nodes for the local-level view. The three terms in the denominator of Equations (14) and (15) denote positive sample pairs, intra-view negative sample pairs, and inter-view negative sample pairs, respectively. We replace the items of the recipe embedding with the user embedding and repeat this process to obtain the contrastive learning losses  $\mathcal{L}_u^g$  and  $\mathcal{L}_u^l$  for the user node embedding computation. So, we can obtain the overall global-level contrast learning losses as follows:

$$\mathcal{L}_{global} = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_r^g + \mathcal{L}_r^l) + \frac{1}{2M} \sum_{i=1}^M (\mathcal{L}_u^g + \mathcal{L}_u^l) \quad (16)$$

Up to now, we completed cross-view self-supervised learning to obtain fine-grained embedding representations of recipe nodes and user nodes.

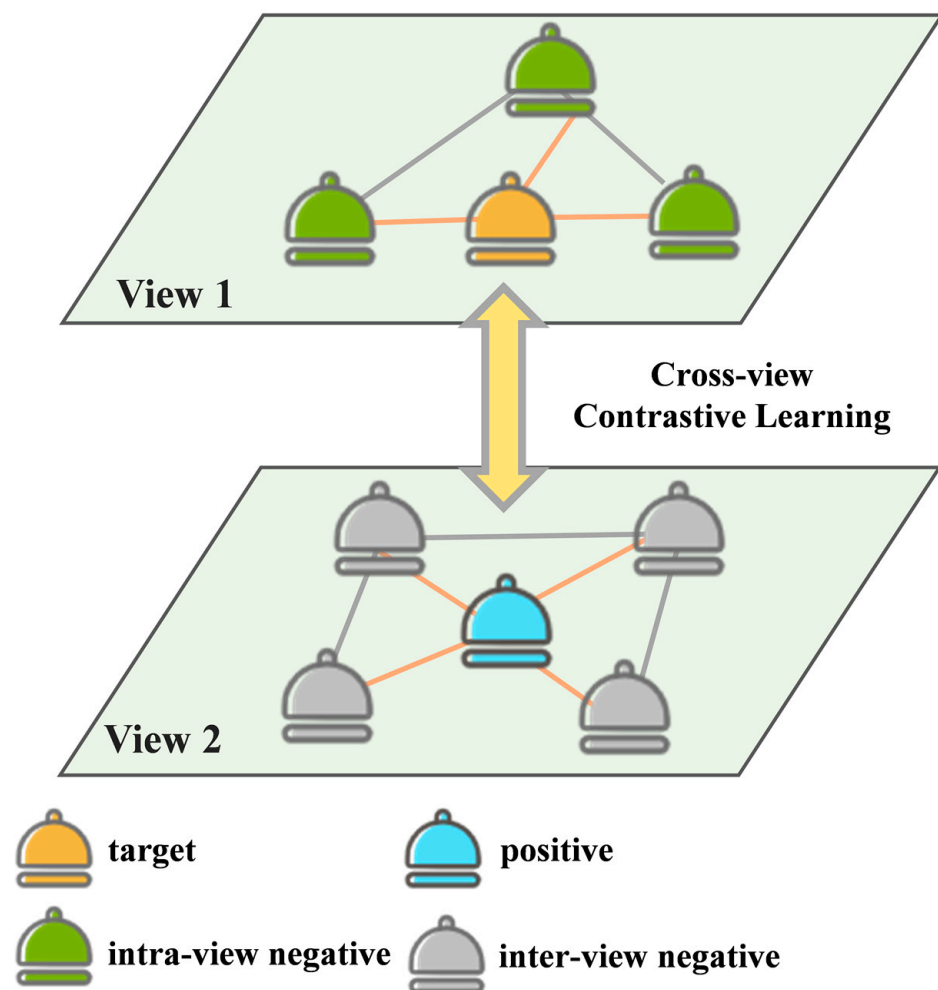


Figure 3. Illustration of cross-view contrastive learning mechanism.

#### 4.3. Predictive Layer and Optimization

In MKCLN, our goal is to recommend recipes to users that match their personalized preferences. Currently, we obtained multiple embedding representations for three views at two levels and optimized them by contrastive learning. Next, we connect these representations by concatenating or summing operation, and the final embedding representation of the node is as follows:

$$\begin{aligned} e_u^* &= e_u^g \parallel e_u^i \\ e_r^* &= e_r^g \parallel (e_r^i + e_r^s) \end{aligned} \quad (17)$$

Finally, we predict the score of the user matching the recipe and make recommendations by an inner product function:

$$\hat{y}(u, r) = e_u^{*\top} e_r^* \quad (18)$$

Heterogeneous graph embedding also suffers from loss, and we combine the self-supervised learning task with the recommendation task by joint training. For recommendation tasks, we usually use Bayesian personalized ranking (BPR) loss [27] for optimization:

$$\mathcal{L}_{BPR} = \sum_{(u,i,j) \in \mathcal{D}_S} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (19)$$

within  $\mathcal{D}_S = \{(u, i, j) \mid (u, i) \in R^+, (u, j) \in R^-\}$  denotes the training set,  $R^+$  denotes the set of recipes that generated (positive) interactions, and  $R^-$  denotes the set of recipes that generated (negative) interactions.  $\sigma(\cdot)$  is the sigmoid function. After that, we combine the local-level and global-level loss with the BPR loss to obtain the MKCLN model integrated loss function as follows:

$$\mathcal{L}_{MKCLN} = \mathcal{L}_{BPR} + \beta \left( \alpha \mathcal{L}_{local} + (1 - \alpha) \mathcal{L}_{global} \right) + \lambda \|\Theta\|_2^2 \quad (20)$$

where  $\alpha$  is the hyperparameter that balances the two contrastive losses and  $\beta$  is the hyperparameter that controls the weight of the contrastive losses.  $\Theta$  is the set of model parameters.

## 5. Experiment

We experimented extensively on two real-world datasets to answer the questions that follow.

RQ1: How does the MKCLN model compare with other recommendation methods for the task of recipe recommendation?

RQ2: Are the individual components of the MKCLN model valid?

RQ3: How do different hyperparameter settings affect the results?

RQ4: Are there differences in performance on the model for recipes with different numbers of interactions (long-tail effects)?

### 5.1. Datasets

We conduct experimental evaluations of the proposed model on the following two real datasets:

- Ta-da (<https://github.com/Eimo-Bai/Ta-da-recipe-dataset> (accessed on 30 June 2022)): The Ta-da dataset is a Chinese recipe dataset constructed by the team of authors, with data from various Chinese recipe sharing social networking websites, specifically containing recipe information, ingredient information, interaction information, taste information, etc. We use the valid recipes and comments information uploaded by users from 2020 to 2022.
- Allrecipes (<https://www.allrecipes.com/> (accessed on 30 June 2022)): This dataset comes from Allrecipes.com, one of the world's largest recipe sharing platforms, and this dataset contains rich content and interactive information, specifically recipe infor-

mation, ingredient information, user interaction information, and image information. We used more than 50,000 recipes from 27 categories published between 2000 and 2018. The final valid data are shown in detail in Table 1.

**Table 1.** The statistics of datasets.

| Dataset      | Ta-da   | Allrecipes |
|--------------|---------|------------|
| users        | 18,679  | 68,768     |
| recipes      | 14,142  | 45,630     |
| ingredients  | 3812    | 29,147     |
| interactions | 246,287 | 1,093,845  |

Our experiments using these two datasets are independent of each other. By pre-processing the datasets, we found that the recipe dataset suffers from a large number of long-tail distribution problems, i.e., most user interactions and comments occur in a small number of recipes, while cold recipes tend to have little interaction information. Therefore, we removed users with less than three interactions in order to filter the noise to ensure that the dataset is not too sparse. In our experiments, we set the ratio of training set, validation set, and test set as 6:2:2.

### 5.2. Baseline

To validate the effectiveness of MKCLN, we compared it with a variety of methods that may be suitable for the recipe recommendation task to explore the applicability of the recommendation model to the recipe recommendation task.

- BPR [27]: A more popular collaborative filtering-based approach that uses a two-by-two matrix decomposition of competing strategies for recommendation and uses Bayesian personalized ranking (BPR) loss optimization.
- CKE [28]: This is a representative embedding-based method that combines text, structure, and visual knowledge in a unified framework for recommendation.
- Metapath2vec [29]: This is a path-based classical recommendation method. For the characteristics of heterogeneous networks, features of meta-paths are extracted to represent the connectivity between users and items.
- FP-MGCN [7]: A GNN-based method, which considers multiple connections of different types of nodes from multiple perspective and uses different propagation methods to enhance the representations in order to obtain high-quality node embeddings for recommendation.
- KGCN [14]: This is a classical GNN-based method that enriches node features by iterative aggregation to propagate the features of items over the knowledge graph, containing structural and semantic information.
- KGAT [17]: An advanced GNN-based method. Using an attention mechanism to aggregate information of neighbor nodes on the user-item-entity graph is more popular in the recommendation field.

### 5.3. Experiments Setup

**Evaluation Metrics.** We evaluate the performance of recipe recommendation in two experimental scenarios: (1) Click-through rate (CTR) prediction. We use AUC and F1 to evaluate the CTR prediction and predict the trained model for each interaction in the test set. (2) Top-K recommendation. The recipes that users did not interact with are ranked and the k highest scoring recipes are recommended to users as items. To evaluate how advanced the model is, we choose recall@k and ndcg@k as evaluation metrics to evaluate the model. All evaluation metrics we report are taken as the average of all results.

**Parameter Settings.** For the MKCLN model and all baseline method implementations, we chose Tensorflow as the experimental platform. To ensure fairness in comparison with

the baseline method, we set the embedding dimension of all models to 64. We optimize our method using Adam [30], and use a grid search for hyper-parameters: the learning range of the regularization parameter is  $[10^{-7}, 10^{-6}, 10^{-5}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$ , the learning rate  $lr$  in the range  $[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05]$ . The other hyperparameters are explored and the optimal values are selected in the following experiments. For the hyperparameter settings in the baseline method, we follow the optimal settings described in the original paper with the default values in the original code.

#### 5.4. Performance Comparison (RQ1)

We conducted comparative experiments on two real recipe datasets and evaluated the overall performance of all methods, as shown in Table 2, Figures 4 and 5. The analysis of the results allows us to draw the following observations:

**Table 2.** Performance of approaches on Ta-da, Allrecipes.

| Methods      | Ta-da  |        | Allrecipes |        |
|--------------|--------|--------|------------|--------|
|              | AUC    | F1     | AUC        | F1     |
| BPR          | 0.8267 | 0.7829 | 0.7337     | 0.6281 |
| CKE          | 0.8389 | 0.7942 | 0.7585     | 0.6384 |
| Metapath2vec | 0.7665 | 0.7405 | 0.6320     | 0.5515 |
| KGCN         | 0.8818 | 0.7980 | 0.7612     | 0.6532 |
| KGAT         | 0.8898 | 0.8177 | 0.7671     | 0.6638 |
| FP-MGCN      | 0.9010 | 0.8225 | 0.7718     | 0.6667 |
| MKCLN        | 0.9184 | 0.8382 | 0.7882     | 0.6798 |

- Our proposed MKCLN model has the best performance in both datasets. The results show that the fine-grained final representations of users and recipes can be better captured through contrastive learning of embedding representations of three views at two levels, so that the final nodes contain more comprehensive information to the extent of having better recommendation results.
- In the baseline methods, CKE performs better than BPR, which proves the superiority of graph data in recommendations. The results of simple embedding using heterogeneous graphs still perform better than the collaborative filtering methods. This is because they do not pay enough attention to the relevant attributes of the recipes.
- The path-based representative method Metapath2vec has the worst performance among all methods. This is because the method relies heavily on manual definition of meta-paths and it is difficult to define the best meta-path. This also shows that structural information is crucial in the embedding of heterogeneous graphs.
- The results of the GNN-based baseline methods KGCN, KGAT, and FP-MGCN outperform other baseline methods, which indicates that the high-quality representations of nodes are very important in recommendation, and the higher-order structural and semantic information in the graph is also very valuable for the potential preferences of users.
- The performance of FP-MGCN is better than KGCN and KGAT because FP-MGCN performs embedding and information propagation for nodes through multiple views separately. This shows that embedded propagation through multiple views is valuable for improving recommendation performance.
- Our proposed MKCLN shows better results than the FP-MGCN in the baseline approach, which illustrates the benefit of self-supervised learning in improving recommendation quality and the importance of the fine-grained representation of nodes on recommendation results.

In the above comparison test, we explore the methods suitable for recipe recommendations. The results of the Ta-da dataset are better than the Allrecipes dataset when compared to the same metric in both datasets. This is because the Ta-da dataset is of higher quality than Allrecipes, which contains recipes from all over the world and also faces some problems with featured ingredients or recipes that have cold attributes and fewer interactions.

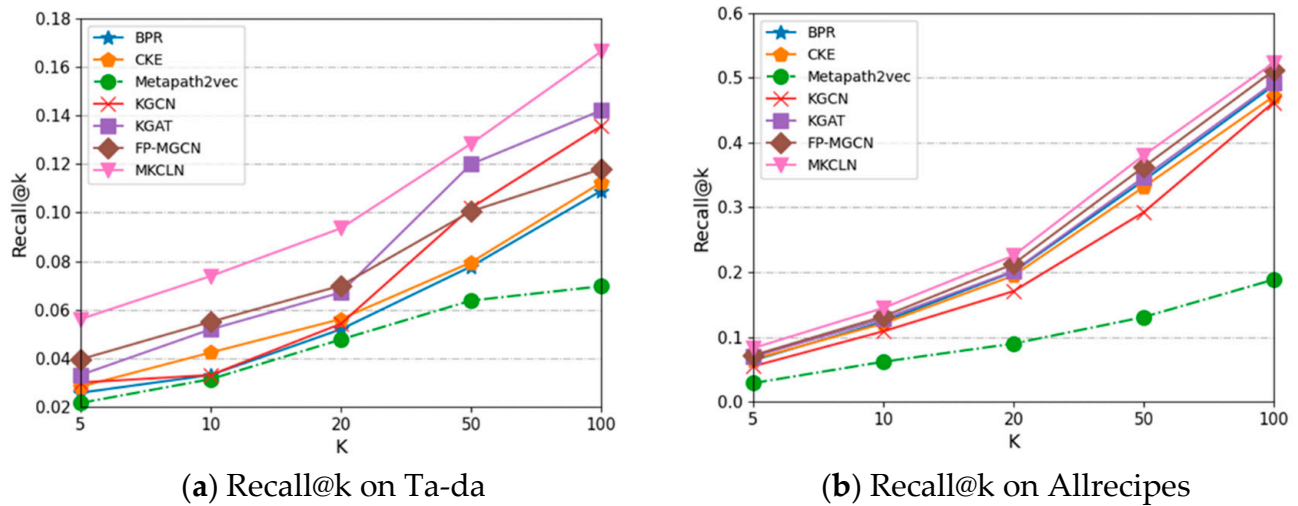


Figure 4. The results of recall@k on two datasets.

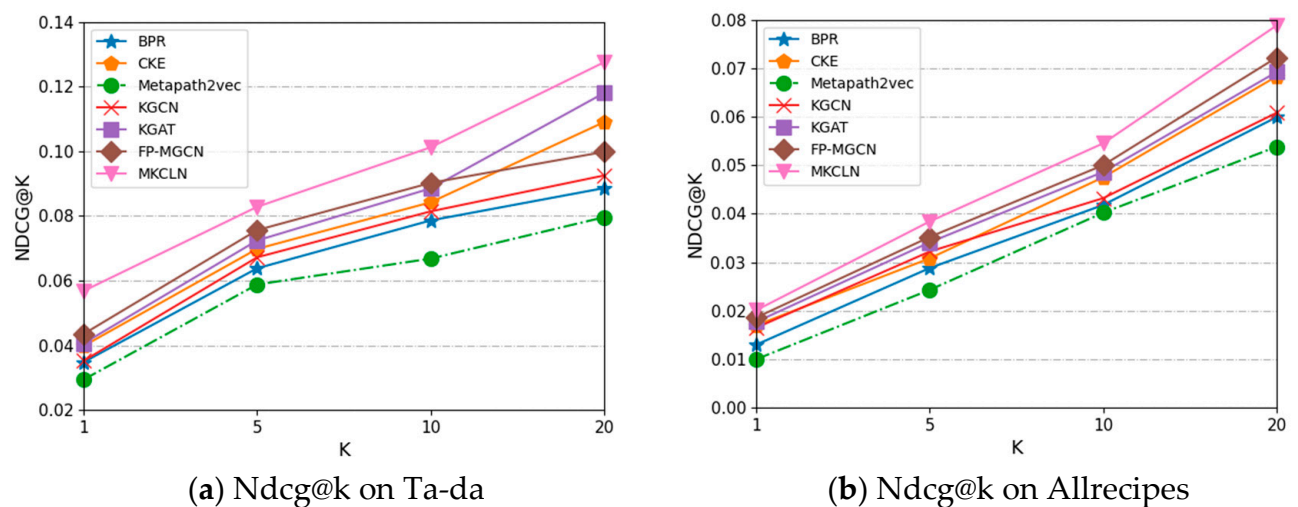


Figure 5. The results of ndcg@k on two datasets.

### 5.5. Ablation Analysis (RQ2)

To verify the effectiveness of each MKCLN component, we conducted ablation experiments on the model, exploring the contribution of a component to the final performance by corrupting a variant of the MKCLN model for the component. We define two variants of the model as MKCLN that corrupts the local-level comparison learning component MKCLN (w/L) and MKCLN that corrupts the global-level comparison learning component MKCLN (w/G), respectively. The experimental results are shown in Table 3, and by observing them we can conclude the following.

**Table 3.** Effect of different component on Ta-da, Allrecipes.

| Model           | Ta-da  |        | Allrecipes |        |
|-----------------|--------|--------|------------|--------|
|                 | AUC    | F1     | AUC        | F1     |
| MKCLN ( $w/L$ ) | 0.8864 | 0.8278 | 0.7828     | 0.6732 |
| MKCLN ( $w/G$ ) | 0.8887 | 0.8133 | 0.7845     | 0.6748 |
| MKCLN           | 0.9184 | 0.8382 | 0.7882     | 0.6798 |

- Removing both local and global levels of contrast learning models significantly degrades the performance, which indicates that collaborative information, semantic information, and high-order structural information have a significant contribution to the recipe recommendation task.
- MKCLN ( $w/L$ ) performs worse than MKCLN ( $w/G$ ). This indicates that for the recipe recommendation task, the collaborative and semantic information contributes more to the fine-grained node representations, and also illustrates the importance of collecting information through two different views.

### 5.6. Study of MKCLN (RQ3)

In order to explore the optimal performance of MKCLN in more detail, we analyze the impact of the parameters on performance through several different perspectives. The exploration of hyperparameters can also reflect the overall strength of the model [31].

#### 5.6.1. Effect of Model Depth

We first explore the MKCLN with different embedding propagation layers, and different embedding layers may lead to different node aggregation results. Specifically, we vary the number of propagation layers  $L$  in the range [1,2,3,4,5], and the results are shown in Table 4, and we can conclude from the analysis that:

- The MKCLN model performs best in the recipe recommendation task at one level of aggregation and slightly decreases in performance at two levels. This indicates that our model collects sufficient feature information by performing aggregation of one or two-hop neighbors, and on the other hand demonstrates the effectiveness of adopting the multiple-view method to collect information.
- The model performance decreases significantly when stacking up to four to five layers. Models that are too deep can lead to overfitting phenomena, while bringing more noise to interfere with the recommended performance.

**Table 4.** The effect of different embedding propagation layers on the model.

|         | Ta-da  |        | Allrecipes |        |
|---------|--------|--------|------------|--------|
|         | AUC    | F1     | AUC        | F1     |
| MKCLN-1 | 0.9184 | 0.8382 | 0.7882     | 0.6798 |
| MKCLN-2 | 0.9180 | 0.8379 | 0.7879     | 0.6796 |
| MKCLN-3 | 0.9162 | 0.8356 | 0.7862     | 0.6768 |
| MKCLN-4 | 0.9125 | 0.8323 | 0.7820     | 0.6734 |
| MKCLN-5 | 0.8876 | 0.8056 | 0.7566     | 0.6486 |

#### 5.6.2. Effect of Loss Function Weigh Parameters

We set two balance hyperparameters  $\alpha$  and  $\beta$  in Equation (20) to control the balance of the two contrast losses and the hyperparameters of the contrast loss weight, respectively. We explore the performance impact of these two hyperparameter settings individually. We vary the parameter  $\alpha$  in the range [0,0.2,0.4,0.6,0.8,1] to explore the effect of local and global proportionality on performance. Afterwards, the optimal  $\alpha$  and varying  $\beta$  is set in the range



[1,0.1,0.01,0.001] to explore the effect of BPR loss and contrastive learning loss weights on performance. The results are shown in Tables 5 and 6, and the following conclusions can be drawn from the observations:

**Table 5.** Effect of parameter  $\alpha$  on model performance.

|                | Ta-da  |        | Allrecipes |        |
|----------------|--------|--------|------------|--------|
|                | AUC    | F1     | AUC        | F1     |
| $\alpha = 0$   | 0.9062 | 0.8305 | 0.7788     | 0.6713 |
| $\alpha = 0.2$ | 0.9184 | 0.8380 | 0.7881     | 0.6796 |
| $\alpha = 0.4$ | 0.9181 | 0.8359 | 0.7880     | 0.6789 |
| $\alpha = 0.6$ | 0.9168 | 0.8344 | 0.7868     | 0.6776 |
| $\alpha = 0.8$ | 0.9160 | 0.8332 | 0.7859     | 0.6768 |
| $\alpha = 1$   | 0.9088 | 0.8297 | 0.7798     | 0.6702 |

**Table 6.** Effect of parameter  $\beta$  on model performance.

|                 | Ta-da  |        | Allrecipes |        |
|-----------------|--------|--------|------------|--------|
|                 | AUC    | F1     | AUC        | F1     |
| $\beta = 1$     | 0.9072 | 0.8326 | 0.7796     | 0.6708 |
| $\beta = 0.1$   | 0.9184 | 0.8382 | 0.7882     | 0.6798 |
| $\beta = 0.01$  | 0.9146 | 0.8352 | 0.7855     | 0.6747 |
| $\beta = 0.001$ | 0.9135 | 0.8328 | 0.7815     | 0.6713 |

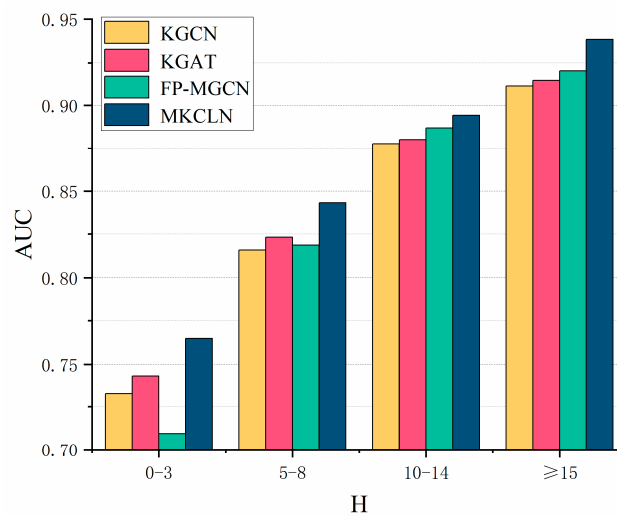
- For the global and local contrastive learning balance parameter  $\alpha$ , the model performance is worst when  $\alpha = 0$  or  $\alpha = 1$ , which indicates that both local-level contrastive learning and global-level are valuable for the results. Meanwhile  $\alpha = 0.2$  when the model achieves optimal performance.
- For the BPR loss and the contrastive learning loss balance parameter  $\beta$ , the model performance is worst when  $\beta = 1$ , which proves the importance of the contrastive learning loss function. The model performs best when  $\beta = 0.1$ .

#### 5.7. User Group Experiment (RQ4)

In the recipe recommendation task, the long-tail phenomenon of recipe dataset tends to affect the performance of the model. The recommendation results for recipes with different popularity can reflect the personalization of the model and also the robustness of the model. We investigate the effect of recipe popularity on the performance of the model by setting four recipe groups with different numbers of interactions (H). We chose to compare the four advanced performance baseline methods in the Ta-da dataset, where the number of recipes in each group is the same. The results are shown in Figure 6 and after analysis we can conclude the following points:

- The MKCLN model is superior in different groups of recipe recommendations, demonstrating that MKCLN is robust and can obtain higher quality embedding representation of nodes compared to other baseline methods.
- For the same method, higher popularity recipes result in higher performance of the model due to the fact that user interaction information enhances the representation of the nodes, i.e., the more historical interactions one has, the more pronounced their features are, thus better capturing the user preference information. The importance of semantic information is also demonstrated.
- We found in exploring the characteristics of recipe content that recipe popularity also affects users' choices, i.e., users prefer more popular dishes to cold ones. So the

recommendation of popular recipes to users that match their preferences does not affect the personalization of the recommendation results.



**Figure 6.** Model performance on recipes with different popularity.

## 6. Conclusions and Future Work

Aiming to solve the problem of the recipe recommendation task containing multiple types of attributes and complex relationships, a multi-level knowledge-aware contrastive learning network (MKCLN) is proposed in this paper. The network is first embedded through three views at global and local levels separately to adequately collect the collaboration information, semantic information, and high-order structure information of the nodes. After that, the final fine-grained representations of users and recipes are obtained by contrastive learning, respectively. Finally, the recipes that satisfy personalized preferences are matched with users through joint training and model prediction functions to complete the recommendation task. We conducted experiments on two recipe datasets containing different countries, and the results show the sophistication and plausibility of the MKCLN model.

In our future work, we will use more evaluation metrics to judge the performance of our proposed model to refine the limitations. In addition, we can also make the following attempts: (1) Try to diversify the recipe recommendations. Currently this work is more concerned with the precision and personalization of recommendation results, which may cause the results not to meet some specific requirements, e.g., weight loss requirements. (2) Try to use more advanced information aggregation methods and message propagation methods for model optimization.

**Author Contributions:** Conceptualization, Z.B.; methodology, Z.B. and S.Z.; software, X.L. and Y.H.; validation, P.L.; resources, Z.B. and P.L.; data curation, Y.C.; writing—original draft preparation, Z.B.; writing—review and editing, Z.B., S.Z. and P.L.; All authors have read and agreed to the published version of the manuscript.

**Funding:** The works described in this paper are supported by The National Natural Science Foundation of China under Grant Nos. 61802352, U1811263; The Program for Young Key Teachers of Henan Province under Grant No. 2021GGJS095; The Project of Science and Technology in Henan Province: 232102210051; The Project of collaborative innovation in Zhengzhou under Grant No. 2021ZDPY0208; The National Natural Science Foundation of China under Grant Nos. 62072414, 62006212, 62102372; The Project of Science and Technology in Henan Province under Grant Nos. 222102210030, 212102210418, 222102210027, 212102210104.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The Allrecipes dataset used in this study is publicly accessible, and the Ta-da dataset is accessible for a partial data sample, to be fully published after the follow-up work Allrecipes dataset is available at: <https://www.allrecipes.com/>, accessed on 30 June 2022. The sample data, a portion of the Ta-da dataset, is available at: <https://github.com/Eimo-Bai/Ta-da-recipe-dataset>, accessed 30 June 2022.

**Acknowledgments:** We sincerely thank the editors and the reviewers for their valuable comments in improving this paper. We would also like to thank L.W. et al. for helping to provide the experimental data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Feng, J.; Xia, Z.; Feng, X.; Peng, J. RBPR: A hybrid model for the new user cold start problem in recommender systems. *Knowl. Based Syst.* **2021**, *214*, 106732. [CrossRef]
2. Wang, W.; Duan, L.-Y.; Jiang, H.; Jing, P.; Song, X.; Nie, L. Market2Dish: Health-aware Food Recommendation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 1–19. [CrossRef]
3. Roshdy, Y.; Hassan, M.A. An Efficient Content-Based Video Recommendation. *J. Comput. Commun.* **2022**, *1*, 48–64. [CrossRef]
4. Trattner, C.; Elsweiler, D.; Howard, S. Estimating the Healthiness of Internet Recipes: A Cross-sectional Study. *Front. Public Health* **2017**, *5*, 16. [CrossRef] [PubMed]
5. Gao, X.; Feng, F.; Huang, H.; Mao, X.-L.; Lan, T.; Chi, Z. Food recommendation with graph convolutional network. *Inf. Sci.* **2021**, *584*, 170–183. [CrossRef]
6. Tahmasebi, F.; Meghdadi, M.; Ahmadian, S.; Valiallahi, K. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. *Multimed. Tools Appl.* **2020**, *80*, 2339–2354. [CrossRef]
7. Zhang, S.; Bai, Z.; Li, P.; Chang, Y. Multi-Graph Convolutional Network for Fine-Grained and Personalized POI Recommendation. *Electronics* **2022**, *11*, 2966. [CrossRef]
8. Chizari, N.; Shoeibi, N.; Moreno-García, M.N. A Comparative Analysis of Bias Amplification in Graph Neural Network Approaches for Recommender Systems. *Electronics* **2022**, *11*, 3301. [CrossRef]
9. Pecune, F.; Callebort, L.; Marsella, S. A recommender system for healthy and personalized recipes recommendations. In Proceedings of the 5th Workshop on Health Recommender Systems Co-Located with ACM RecSys, Rio de Janeiro, Brazil, 26 September 2020; pp. 15–20.
10. Vivek, M.B.; Manju, N.; Vijay, M.B. Machine learning based food recipe recommendation system. In *Proceedings of the International Conference on Cognition and Recognition*; Springer: Singapore, 2018; pp. 11–19.
11. Khan, M.A.; Rushe, E.; Smyth, B.; Coyle, D. Personalized, health-aware recipe recommendation: An ensemble topic modeling based approach. *arXiv* **2019**, arXiv:1908.00148. [CrossRef]
12. Gao, X.; Feng, F.; He, X.; Huang, H.; Guan, X.; Feng, C.; Ming, Z.; Chua, T.-S. Hierarchical Attention Network for Visually-Aware Food Recommendation. *IEEE Trans. Multimed.* **2019**, *22*, 1647–1659. [CrossRef]
13. Tian, Y.; Zhang, C.; Guo, Z.; Huang, C.; Metoyer, R.; Chawla, N.V. RecipeRec: A Heterogeneous Graph Learning Model for Recipe Recommendation. *arXiv* **2022**, arXiv:2205.14005. [CrossRef]
14. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313. [CrossRef]
15. Ma, X.; Wang, R. Personalized Scientific Paper Recommendation Based on Heterogeneous Graph Representation. *IEEE Access* **2019**, *7*, 79887–79894. [CrossRef]
16. 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, Virtual Event, 25–30 July 2020; pp. 639–648.
17. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 July 2019; pp. 950–958.
18. Schneider, D.; Sarfraz, S.; Roitberg, A.; Stiefelwagen, R. Pose-based contrastive learning for domain agnostic activity representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 3433–3443.
19. Chen, Q.; Lacomis, J.; Schwartz, E.J.; Neubig, G.; Vasilescu, B.; Goues, C.L. Varclr: Variable semantic representation pre-training via contrastive learning. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 8–20 May 2022; pp. 2327–2339.
20. Dave, I.; Gupta, R.; Rizve, M.N.; Shah, M. TCLR: Temporal contrastive learning for video representation. *Comput. Vis. Image Underst.* **2022**, *219*, 103406. [CrossRef]
21. Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; Xie, X. Self-supervised Graph Learning for Recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 11–15 July 2021; pp. 726–735. [CrossRef]

22. Wang, X.; Liu, N.; Han, H.; Shi, C. Self-supervised heterogeneous graph neural network with co-contrastive learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 14–18 August 2021; pp. 1726–1736.
23. Dang, C.N.; Moreno-García, M.N.; De la Prieta, F. An Approach to Integrating Sentiment Analysis into Recommender Systems. *Sensors* **2021**, *21*, 5666. [[CrossRef](#)] [[PubMed](#)]
24. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907. [[CrossRef](#)]
25. Zou, D.; Wei, W.; Mao, X.-L.; Wang, Z.; Qiu, M.; Zhu, F.; Cao, X. Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System. *arXiv* **2022**, arXiv:2204.08807. [[CrossRef](#)]
26. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Graph Contrastive Learning with Adaptive Augmentation. In Proceedings of the Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080.
27. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
28. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
29. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [[CrossRef](#)]
31. Li, X.; Jiang, W.; Chen, W.; Wu, J.; Wang, G.; Li, K. Directional and Explainable Serendipity Recommendation. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 122–132. [[CrossRef](#)]