Self-supervised Calorie-aware Heterogeneous Graph Networks for Food Recommendation

YAGUANG SONG, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA); School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), China
XIAOSHAN YANG and CHANGSHENG XU, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA); School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), China and Peng Cheng Laboratory, China

With the rapid development of online recipe sharing platforms, food recommendation is emerging as an important application. Although recent studies have made great progress on food recommendation, they have two shortcomings that are likely to affect the recommendation performance. (1) The relations between ingredients are not considered, which may lead to sub-optimal representations of recipes and further result in the neglect of the user's personalized ingredient combination preference. (2) Existing methods do not consider the impact of users' preferences on calories in users' food decision-making process. In this article, we propose a Self-supervised Calorie-aware Heterogeneous Graph Network (SCHGN) to model the relations between ingredients and incorporate calories of food simultaneously. Specifically, we first incorporate users, recipes, ingredients, and calories into a heterogeneous graph and explicitly present the complex relations among them with directed edges. Then, we explore the co-occurrence relation of ingredients in different recipes via self-supervised ingredient prediction. To capture users' dynamic preferences on calories of food, we learn calorie-aware user representations by hierarchical message passing and compute a comprehensive user-guided recipe representation by attention mechanism. The final food recommendation is accomplished based on the similarity between a user's calorie-aware representation and the user-guided representation of a recipe. Extensive experiment results on benchmark datasets demonstrate the effectiveness of the proposed method.

CCS Concepts: • **Information systems → Recommender systems;**

Additional Key Words and Phrases: Food recommendation, recipe calories, heterogeneous graph, self-supervised learning

This work was supported by National Key Research and Development Program of China (No. 2018AAA0100604), National Natural Science Foundation of China (Nos. 61720106006, 62036012, 62072455, 61721004, U1836220, U1705262, 61872424), Key Research Program of Frontier Sciences of CAS (QYZDJ-SSW-JSC039), Beijing Natural Science Foundation (L201001).
Authors' addresses: Y. Song, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA); School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), NO. 95 Zhongguancun East Rd, Haidian District, Beijing, China, 100190; email: songyaguang2019@ia.ac.cn; X. Yang and C. Xu, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA); School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), NO. 95 Zhongguancun East Rd, Haidian District, Beijing, China, 100190, and Peng Cheng Laboratory, Shenzhen, China; emails: {xiaoshan.yang, csxu}@nlpr.ia.ac.cn.

1551-6857/2023/01-ART27 $15.00
https://doi.org/10.1145/3524618

ACM Reference format:
Yaguang Song, Xiaoshan Yang, and Changsheng Xu. 2023. Self-supervised Calorie-aware Heterogeneous Graph Networks for Food Recommendation. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s, Article 27 (January 2023), 23 pages.
https://doi.org/10.1145/3524618

1 INTRODUCTION

Food plays an essential role in everyone's life. With the rapid development of online sharing platforms such as yummly[1] and social media, people tend to share their daily lives including their diets. In the meantime, compared with the past, people nowadays have more choices on their daily diets. To provide suitable food or diets for people, food recommendation [42] is emerging as an important application, which attracts increasing attention in both industry and academia.

The task of recommendation has been widely studied in the literature [1, 24, 26, 32, 46, 66, 71], which generally utilizes the past interaction records to infer the user's preference and recommend items. Compared with the general recommendation [23, 24], which largely focuses on the domains of E-commerce products or movies, food recommendation has its special characteristics [13, 42]. Specifically, food content information such as ingredients, cooking methods, and visual appearance greatly influence whether a user will choose a recipe or not [11, 13, 36, 39, 53]. These features make it difficult to infer the user's complex preferences purely from the user-recipe interactions. Typically, recent studies on food recommendation begin to focus on modeling the impact of various characteristics of food on the user's decision-making process [13, 39]. Gao et al. [13] propose a food recommender system incorporating user-recipe interaction history, ingredients, and food images to model the user's preference and contribute a large-scale food recommendation dataset. This food recommender system adopts neural networks and attention mechanism to jointly model the user's preference on different ingredients and aspects of food in a hierarchical manner. Following Reference [13], to model the user's personalized visual preference, Meng et al. [39] propose a new food recommendation framework based on multi-task learning paradigm to learn the visual features of food images that can fuse both the semantic and personalized visual information.

Although the existing methods have taken various factors of food into account in the recommendation process and achieved promising results, there are two major shortcomings that are likely to affect the recommendation performance: (1) The relations (e.g., co-occurrence) between ingredients are not considered, which may lead to sub-optimal representations of recipes. More importantly, users' preferences for different combinations of ingredients will not be well captured due to the neglect of the relations between ingredients. For example, as illustrated in Figure 1, some ingredients often appear together in recipes selected by User A (denoted by boxes with solid line), which reflects the user's personalized ingredient combination preference. Since ingredients are the building blocks of a recipe, it is important to emphasize the interactions among the ingredients in learning an effective representation of the recipe and modeling the user's preference for ingredient combinations. (2) The existing methods do not consider the impact of users' preference on calories of food in users' food decision-making process. Although previous studies [9, 15, 56, 61, 68] on health-aware food recommendation have considered nutrition information such as calories during recommendation, most of them utilize calories as rules or conditions to recommend healthier food. Different from them, we believe that calories is one of the aspects that has influence on the user's food preference and we also need to consider it for more effective recommendation. We present an example in Figure 1, where we sample two users and show their recent food choices. The same

[1] https://www.yummly.com/.

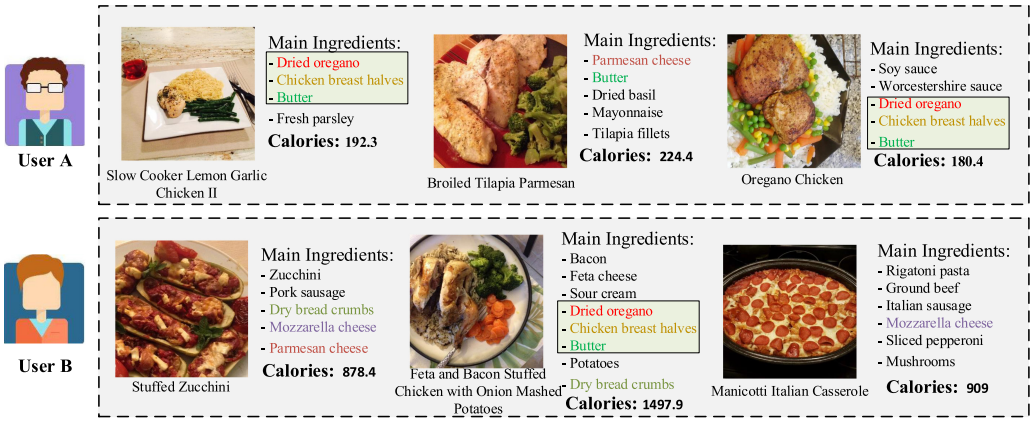

Fig. 1. Illustration of users' preference on ingredient combinations (i.e., some ingredients, such as "*dried oregano*"+"*chicken breast halves*"+"*butter*," appear together in different recipes selected by User A) and users' preference on calories of recipes (i.e., the recipes chosen by User A are usually low calorie, while User B is more likely to choose recipes with higher calories).

ingredients are marked with the same colors. It can be seen that there are overlapping ingredients in the two users' interacted recipes, which also makes the appearance of dishes look similar. However, the recipes chosen by User A are usually low-calorie, while User B is more likely to choose recipes with higher calories, which shows the calorie preferences of two users. Therefore, not only the user's taste or visual preference of recipes but also the user's preference on calories play a critical role in the user's food decision-making process. The taste of food is related to both the combination of ingredients and the cooking method. In this article, the "taste" mainly depends on the combination of ingredients.

The above issues that seem easy to solve are not trivial. **(1)** To capture the relations between ingredients, we can simply utilize graph neural networks [30, 35] or self-attention methods [8, 59], which are widely adopted for modeling structured data. However, the scarcity of supervised signals of user-item pairs in the food recommendation task is likely to lead to the ineffectiveness of modeling the relations between ingredients. **(2)** For the calorie issue, we can easily infer a user's personalized awareness from the user's historical data and define a calorie-related user profile. However, such a solution ignores the dynamic impacts of calorie and other factors on the user's diet choice. For example, a user may prefer low-calorie food in general, but when a recipe with high calories contains his/her favorite ingredients, he/she may still choose it.

To comprehensively explore the impact of the user's preference on combinations of ingredients and calorie information, in this article, we propose a **Self-supervised Calorie-aware Heterogeneous Graph Network (SCHGN)** where we aim to recommend suitable food for the target user, given the user-recipe interactions, ingredients, images, and calories of food. Specifically, we first incorporate users, recipes, ingredients, and calories into a heterogeneous graph and use directed edges to explicitly present the complex relations among them. Then, to effectively model the relations between ingredients to help capture the user's personalized ingredient combination preference, we adopt the idea of self-supervised learning [8, 25, 27, 43, 50], which is a newly emerging paradigm aiming to let the model learn from the intrinsic structure of the raw data and explore the co-occurrence of ingredients in different recipes via self-supervised ingredient prediction. Meanwhile, to capture users' awareness of calories based on historical recipes, the proposed method learns calorie-aware user representations by hierarchically aggregating useful signals from

representations of ingredient, calorie, and recipe nodes in the heterogeneous graph. Next, to dynamically explore a user's preference for ingredients, appearance, or calories of a specific recipe, we learn a comprehensive user-guided recipe representation through attention networks based on both the calorie-aware user representation and the multi-modal information of the recipe. Finally, the food recommendation is accomplished based on the similarity between a user's calorie-aware representation and the user-guided representation of a recipe.

Our contributions are summarized as follows:

- We propose a **Self-supervised Calorie-aware Heterogeneous Graph Network (SCHGN)** for food recommendation, which can model the relations of ingredients and capture the user's preference on food calories simultaneously.
- To effectively model the complex relations between ingredients, we adopt the idea of self-supervised learning and explore the co-occurrence of ingredients in different recipes via self-supervised ingredient prediction.
- We highlight the significance of the user's preference on calories in food decision-making process and propose to explicitly integrate calories of food with hierarchical message passing to dynamically explore a user's preference of taste and calories for a specific recipe.
- We validate the effectiveness of our proposed model on the benchmark dataset [13]. Extensive experimental results demonstrate that our model achieves superior performance against the existing methods.

We organize the remainder of this article as follows: In Section 2, we review the related work. Section 3 describes the details of our proposed method. In Section 4, the experimental results and analysis are given on the benchmark dataset. The conclusion and future work are presented in Section 5.

2 RELATED WORK

In this section, we review the most related work to our method in food recommendation, graph neural networks, and self-supervised learning.

2.1 Food Recommendation

Food recommendation has received more and more attention in recent years. In general, food recommendation aims to provide a list of ranked food items for users to meet their personalized needs [42, 55, 58, 61].

Some of the existing studies adopt the user's ratings of recipes or past interactions to build food recommender systems [11, 14, 19, 56] based on collaborative filtering framework, which is widely adopted in recommendation [23, 33, 45, 51, 72]. Harvey et al. [19] utilize SVD for rating matrix factorization and obtain better performance than baselines in Reference [11]. Ge et al. [14] propose a matrix factorization method that leverages latent factors and user supplied tags for food recommender systems to achieve significantly better accuracy than standard matrix factorization methods. Trattner et al. [56] conduct a comprehensive experiment on testing a diverse range of collaborative filtering methods using a large dataset and demonstrate the superiority of **Latent Dirichlet Allocation (LDA)** and weighted matrix factorization.

In addition to relying solely on the user's interaction with the recipes, the content information of recipes, such as ingredients and dish images, is also crucial for modeling the user's recipe preference [11–13, 39, 53, 67, 68]. Exploiting the content information of recipes can alleviate the problem of data sparsity as well. Early studies focus on exploiting the ingredients of the recipe to make recommendation [11, 12, 53]. Then, some researchers [10, 67, 68] propose to utilize the image of recipes to enhance the food recommender system. In recent years, to comprehensively model

the user's food preference, Gao et al. [13] propose a food recommendation framework, exploiting a neural network to model the interaction between users and recipes, the images of recipes, and the ingredients simultaneously. This method adopts a pre-trained recipe classification model to extract image features, which only contain semantic information such as ingredients, and cannot capture user's personalized visual preferences. Hence, Meng et al. [39] propose a visually aware food recommendation framework that jointly optimizes the image feature extractor with the recommendation task and the recipe ingredient classification task making the learned image features contain both the semantic information and the personalized visual information. It is worth noting that this method adopts a different setting from ours, where only users' interactions and recipe images are taken as input and the pre-processed ingredients are used as the supervision information. For the knowledge graph-based food recommendation method [5], Chen et al. propose to formulate the personalized food recommendation as a constrained question answering over a food knowledge graph named FoodKG.[2] Besides the user query, they take the dietary preferences and nutrition information as additional constraints to retrieve recipe from the food knowledge graph. To sum up, none of the these methods consider the user's preference on calories of food, and the relations of ingredients are not well exploited, which limits the recommendation performance.

Health-aware food recommendation, as a special domain of food recommendation, has also been explored with efforts [2, 42, 52, 54, 61], since diet itself is closely related to people's health. In fact, some previous studies on health-aware food recommendation have incorporated calorie information into the recommendation process [9, 15, 56, 68], but most of them aim to build a health-driven recommender system, which balances the user's preference and nutrition factors of recipes from the perspective of tradeoff. Haussmann et al. [20] build a food knowledge graph, including recipes, ingredients, and nutrition information as entities and propose a knowledge-based question answering application. Different from the existing methods, we model the impact of calorie factor on the user's decision-making implicitly utilizing the user's interaction records and the calories of food. The goal of our work is to make the recommendation results more accurate. We think the proposed calorie-aware food recommendation system is useful and necessary on a recipe-sharing platform where the users always rationally choose food. If a user has chosen many high-calorie foods on the platform, then he might be thin or suffering from malnutrition and want to gain weight. In this case, the system needs to recommend foods with high calories rather than low-calorie ones. If a user is on a diet, then the system can capture the user's calorie preference and recommend low-calorie food.

2.2 GNN for Recommendation

Graph neural network (GNN) can capture the dependence of nodes in graphs via message passing. The main idea of GNN is to distill useful information from neighbors and combine the aggregated information to update the node representation. In recent years, GNNs have attracted much attention and have been widely applied in many fields due to the great expressive power. Early studies on graph convolution aim to define the convolution kernel in spectral domain [3, 6], which is computationally expensive and lacks generalization. Kipf et al. [30] propose a simplified and widely used paradigm of **graph convolutional networks (GCNs)**. GraphSage [18] designs a permutation-invariant aggregator for message passing in spatial domain. Veličković et al. [60] introduce the multi-head attention mechanism into the message passing between nodes of the graph. Motivated by the strength of graph neural networks, recent studies [35, 62, 63, 69, 74] adapt GCN to model the complex relationships between users, items, and other properties for recommendation. Li et al. [35] propose a hierarchical graph to model the relationship among users, items, and outfits

[2] https://foodkg.github.io/index.html.

and adopt GNNs to obtain more expressive representations for outfit recommendation. Zheng et al. [74] propose a graph-based solution to unify the influence of item price and category, where graph neural networks are utilized to learn price-aware and category-dependent user representations. In this article, for the first time, we apply GNNs to personalized food recommendation and build a calorie-aware heterogeneous graph to explicitly explore the complex relationships among users, recipes, ingredients, and calories. We utilize message propagation in GNNs to learn calorie-aware user representations and dynamically explore users' preference of taste and calorie for recipes.

2.3 Self-Supervised Learning

Self-Supervised Learning (SSL) has gained increasing attention [8, 25, 41] recently, which aims at learning representations on an auxiliary objective where the supervision information is obtained from the raw data. Self-supervised learning has been employed for learning representations in multiple areas.

In natural language processing, self-supervised learning tasks have been widely explored for predicting the adjacent words [41] or predicting the next sentence [8] given the previous sequences. Several self-supervised objectives have been introduced in computer vision community [31] as well, including: (i) predicting image rotations [16]; (ii) predicting relative path locations [44]; (iii) predicting next video frames [49]; (iv) minimizing the similarity of images with augmentations [4], and so on.

In recommendation, Zhou et al. [75] propose a self-supervised learning method based on mutual information maximization [25] for sequential recommendation and design four self-supervision tasks for pre-training. Ma et al. [37] extract additional supervision signals by investigating the longer-term future instead of just the next immediate behavior and perform the self-supervision in latent space. Xin et al. [64] show that combining SSL with reinforcement learning is effective to capture long-term user interest in sequential recommendation.

In food computing, Lee et al. [17] introduce an online recipe generation system for cooking recipe generation, where the generator comprises a generative pre-trained language model GPT-2 fine-tuned on a large cooking recipe dataset. The recipe generation system can generate cooking instructions according to given recipe title and ingredient texts and generate ingredients given recipe title and cooking instruction texts. Li et al. [34] propose a joint approach to learn pretrained recipe representations by considering the alignment of ingredients and cooking instructions in latent space as the supervision. At present, existing methods in personalized food recommendation do not consider the co-occurrence of ingredients in different recipes. In this work, we apply self-supervised learning to food recommendation and aim to model the relations between ingredients by reconstructing the masked ingredients given the other ingredients of the recipe. Tsukuda et al. [57] propose a method for recipe search by adding and removing ingredients based on the co-occurrence probability of ingredient pairs. Yokoi et al. [70] focus on arrangement of ingredients and propose a framework for typicality analysis of the combination of ingredients. Although they take the relation between ingredients into consideration, they focus on recipe retrieval rather than personalized food recommendation. Marin et al. [38] introduce Recipe1M+, a large-scale recipe dataset, and propose a multimodal recipe retrieval method based on joint neural embedding with semantic regularization. They adopt word2vec [40] to encode the ingredient texts and then use LSTM to obtain the final representation of combined ingredients. However, the word2vec is pretrained on public text corpora. Therefore, this method can only capture the word co-occurrence in semantic space. Compared with their method, we explicitly consider the co-occurrence of ingredients by performing self-supervised ingredient prediction on recipe data, which can not only reflect the semantic co-occurrence of different ingredients but also reflect the common cooking styles or eating habits. Although Salvador et al. [48] also adopt self-supervised learning, the purpose is to

model the relation between ingredients and cooking instructions of the recipe, rather than the relation between ingredients. Besides, the above two methods focus on cross-modal recipe retrieval rather than personalized food recommendation.

3 METHODOLOGY

3.1 Problem Formulation

The aim of food recommendation is to predict the user's preference over recipes. In this article, we contribute to incorporating the calorie factor of recipes to improve the performance of food recommendation. Therefore, we reformulate the food recommendation task as follows:

Before giving the formal problem definition, we first introduce the following important notations: We use U and I to denote a set of users and a set of recipes, respectively. The user-recipe interaction matrix is denoted as $\mathbf{Y} \in \mathbb{R}^{N_U \times N_I}$, where N_U and N_I are the number of users and recipes. For a user u and recipe i, an entry $y_{ui} = 1$ means that the user has interacted with the recipe. Besides, the recipe i is equipped with an image feature $\mathbf{v}_i \in \mathbb{R}^{2048}$, a set of ingredients $\mathbf{g}_i \in \mathbb{R}^{N_K}$, and a calorie factor c_i. $\mathbf{g}_i$ is a multi-hot encoding vector with $g_i^k = 1$ denoting that the recipe i contains the ingredient k. N_K is the total number of ingredients occurred in I. The calorie factor $c_i \in \mathbb{R}^{N_C}$ is a one-hot vector denoting the calorie level of the recipe i, where N_C is the number of calorie levels. Based on the above notations, the food recommendation task is formally defined as:

Input: User-recipe interaction matrix $\mathbf{Y}$, recipe ingredients $[\mathbf{g_1}, \ldots, \mathbf{g}_{N_I}]$, recipe image features $[\mathbf{v}_1, \ldots, \mathbf{v}_{N_I}]$, and recipe calories $[c_1, \ldots, c_{N_I}]$.

Output: An interaction function $y_{ui} = f(u, i, \mathbf{g}_i, \mathbf{v}_i, c_i)$, which predicts the estimated probability that user u would interact with recipe i.

3.2 Overview

In this article, we propose a **self-supervised calorie-aware heterogeneous graph network (SCHGN)** for food recommendation. Figure 2(a) shows the overview architecture of the proposed SCHGN. Given a user-recipe pair (u, i) where the recipe has ingredients $\mathbf{g}_i$, image feature $\mathbf{v}_i$, and calorie factor c_i, the model aims to predict the probability that the user u will interact with the recipe i. The proposed SCHGN consists of five main modules. (1) We build a heterogeneous graph to explicitly explore the complex relationships of users, recipes, ingredients, and calories. (2) We use the self-supervised ingredient prediction module (details are shown in Figure 2(b)) to model the relationships between ingredients by predicting the masked ingredients given the ingredient set $\mathbf{g}_i$ of recipe i. (3) We learn calorie-aware user representations through message propagation from graph neural networks. (4) To dynamically explore the user's preference of taste and calories for the recipe, we learn a comprehensive user-guided recipe representation through attention networks. (5) We utilize the calorie-aware user representation and the user-guided recipe representation to accomplish the food recommendation task. The overall framework of our method can be jointly optimized with the self-supervised ingredient prediction loss and the food recommendation loss in an end-to-end manner. More details of the five main modules will be illustrated as follows.

3.3 Heterogeneous Graph

Given the user-recipe interactions where each recipe has ingredients and calorie factor, it is challenging to explicitly model the complex relations among users, recipes, ingredients, and calories, and further capture users' calorie awareness, since users are not directly related to the calories of recipes. To address this challenge, we organize users, recipes, ingredients, and calories into a heterogeneous graph. For example, if a recipe i contains the ingredient g and its calorie attribute is c, then an ingredient node and a calorie node will be connected to the recipe node. Similarly, if

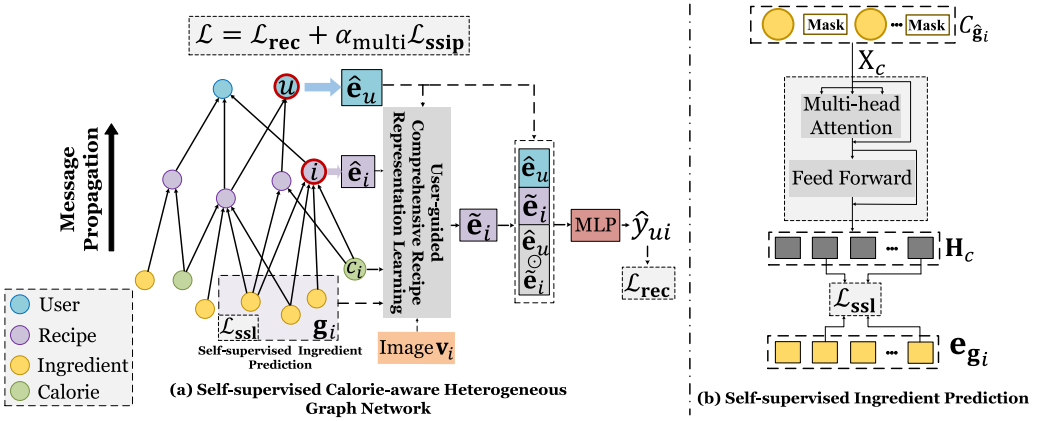

Fig. 2. Overview of the proposed self-supervised calorie-aware heterogeneous graph network **(a)** and the self-supervised ingredient prediction module **(b)**. To predict whether a user u will interact with a recipe i, we first build a heterogeneous graph consisting of users, recipes, ingredients, and calories. Then, the self-supervised ingredient prediction is adopted to model the relationships between ingredients. Next, we learn a calorie-aware user representation $\hat{\mathbf{e}}_u$ by hierarchically aggregating useful signals from representations of ingredient, calorie, and recipe nodes. Finally, we learn a comprehensive user-guided recipe representation $\tilde{\mathbf{e}}_i$ through attention networks for the final prediction $\hat{y}_{ui}$. For the self-supervised ingredient prediction, we aim to recover the masked ingredients given the surrounding context ingredients $\mathbf{C}_{\hat{\mathbf{g}}_i}$ with a multi-head self-attention architecture.

a user u has chosen recipe i, then there is an edge from the recipe node to the user node. Then, we can leverage the connectivity of graph nodes to explore underlying relationships by propagating embedding vectors from calories to users with recipes as the bridge.

To be more specific, there are four types of nodes—users, recipes, ingredients, and calories, which basically can be divided into three levels—users, recipes, and recipe attributes (i.e., ingredients and calories). For edges, we connect the attribute nodes to the recipes if the recipe i includes the ingredient g or belongs to the calorie level c. Existing work [28, 32] has shown that a user's preference can be reflected by personal history directly, and similar users would have similar preferences on recipes. Therefore, we connect the recipes to the users if user u has interacted with recipe i. Figure 2(a) shows an example of the built heterogeneous graph. Such a heterogeneous graph highlights the connections cross levels.

Embedding Initialization. Since the user ID, recipe ID, ingredient ID, and calorie ID are encoded as one-hot vectors, to characterize the latent features, we represent each user/recipe/ingredient/calorie ID with an embedding representation. That is, we represent each node with a separate embedding $\mathbf{e}_* \in \mathbb{R}^d$ where $*$ refers to the node and d is the embedding size. We adopt the same embedding method as in Reference [13]. For example, given the one-hot encoding of a specific user u, denoted as $O_u \in \mathbb{R}^{N_U}$, we can project it into an embedding as follows:

$$e_u = \mathbf{E}_U^T O_u, \tag{1}$$

where the $e_u \in \mathbb{R}^d$ is the embedding of user u. $\mathbf{E}_U \in \mathbb{R}^{N_U \times d}$ is a learnable embedding matrix of all users. We can obtain other node embeddings in a similar way. As a result, we maintain an embedding matrix for all the nodes denoted as $\mathbf{E} \in \mathbb{R}^{(N_U+N_I+N_K+N_C)\times d}$, which is composed of the embeddings of users, recipes, ingredients, and calories. Here, N_U, N_I, N_K, and N_C are the number of the users, recipes, ingredients, and calorie levels, respectively.

3.4 Self-supervised Ingredient Prediction

The relations between ingredients (e.g., co-occurrence) are not considered in previous studies on food recommendation, which easily leads to sub-optimal representations of recipes and further results in the neglect of the personalized ingredient combination preference. Due to the scarcity of supervision information, inspired by the masked language model such as BERT [8], we propose to model the relationship between ingredients in a self-supervised learning manner as a Cloze task. The Cloze setting is described as below: Given the ingredient set of a recipe, at each training step, a proportion of ingredients in the set is randomly masked (i.e., replaced with "[mask]" token). Then, we predict the masked ingredients based on the other ingredients of the recipe with self-attention [59, 60], which has shown its power on quantifying the interdependence among the inputs.

More specifically, as illustrated in Figure 2(b), given the original ingredient set $\mathbf{g}_i$ of a recipe i with L ingredients and their corresponding representations collected from $\mathbf{E}$ denoted as $\mathbf{X}_{\mathbf{g}_i} \in \mathbb{R}^{L\times d}$, we randomly mask M ingredients $\hat{\mathbf{g}}_i$ by replacing them with the "mask" token and treat the obtained set as the surrounding context set $\mathbf{C}_{\hat{\mathbf{g}}_i}$. We obtain the embedding $\mathbf{X}_t \in \mathbb{R}^{L\times d}$ of $\mathbf{C}_{\hat{\mathbf{g}}_i}$ by collecting the representations of ingredients from the embedding matrix $\mathbf{E}$ and utilizing a specific embedding for the "mask" token, which is randomly initialized. To capture the co-occurrence information of ingredients, we first apply a multi-head self-attention layer to $\mathbf{X}_t$:

$$\begin{aligned} \mathbf{Z} &= \text{MultiHead}(\mathbf{X}_t) = \text{Concat}(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_h)\mathbf{W}^O \\ \text{where } \mathbf{Z}_r &= \text{Att}\left(\mathbf{X}_t\mathbf{W}_r^Q, \mathbf{X}_t\mathbf{W}_r^K, \mathbf{X}_t\mathbf{W}_r^V\right). \end{aligned} \tag{2}$$

Here, $\mathbf{W}_r^Q$, $\mathbf{W}_r^K$, $\mathbf{W}_r^V$ are the corresponding learnable parameters for each attention head. $\mathbf{W}^O$ is the parameter of the projection layer applied to the concatenated matrix. h denotes the number of heads. For the multi-head self-attention, we adopt the same architecture proposed in Reference [59], which adopts multiple scaled dot-product operations with unshared learnable transformation matrices to process the input data and capture interactive relation information in multiple projection spaces. Att denotes a function that computes the attention scores using scaled dot-product:

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_t}}\right)\mathbf{V}, \tag{3}$$

where $\sqrt{d_t}$ is the scale factor to avoid large values of the inner product. We denote the output of the multi-head self-attention layer as $\mathbf{Z} \in \mathbb{R}^{L\times d}$.

In addition to the multi-head self-attention layer, following Reference [59], we adopt a point-wise feed forward network consisting of two linear transformations with a ReLU activation in between, applied to each position of the input set, which is formally defined as:

$$\begin{aligned} \mathbf{Z}' &= [\text{FFN}(\mathbf{Z}[1,:]); \ldots; \text{FFN}(\mathbf{Z}[L,:])], \\ \text{FFN}(x) &= \left(\text{ReLU}(x\mathbf{W}_{f1} + \mathbf{b}_{f1})\right)\mathbf{W}_{f2} + \mathbf{b}_{f2}, \end{aligned} \tag{4}$$

where $\mathbf{Z}' \in \mathbb{R}^{L\times d}$ is the output of one self-attention layer and feed forward network, and $\mathbf{W}_{f1}$, $\mathbf{b}_{f1}$, $\mathbf{W}_{f2}$, $\mathbf{b}_{f2}$ are trainable parameters.

We can stack more multi-head self-attention and feed forward layers. Then, we obtain the final output representation denoted as $\mathbf{H}_t \in \mathbb{R}^{L\times d}$, which encodes the context information (i.e., co-occurrence) of the given ingredients. To constrain the model, we adopt a self-supervised learning loss based on mutual information maximization [25] to maximize the similarity between the original embeddings of masked ingredients collected from $\mathbf{X}_{\mathbf{g}_i}$ and the corresponding predicted context embeddings collected from $\mathbf{H}_t$. More details of the self-supervised learning loss will be introduced

in Section 3.8. With the self-supervised ingredient prediction, the ingredient embeddings in $\mathbf{E}$ will be continuously updated and the relations between ingredients can be well captured, which will lead to more effective recipe representations.

3.5 Calorie-aware User Representation Learning

In recent years, graph neural networks [30] have made rapid progress and have been applied in many tasks. Several recent studies on graph networks [60, 65] have shown that the information propagation over graph structure is able to effectively extract useful information from multi-hop neighbors and encode the connection information into the node representation. Upon our heterogeneous graph, to capture users' awareness of calories based on historical recipes, we learn calorie-aware user representations by hierarchically aggregating useful signals from representations of ingredient, calorie, and recipe nodes.

3.5.1 Message Propagation. Specifically, suppose the node n_i and its neighboring node n_j are two connected nodes in our heterogeneous graph. The information being propagated from the neighboring node n_j to the node n_i is formalized as:

$$\mathbf{m}_{n_j \to n_i} = \frac{1}{|\mathcal{N}_{n_i}|}(\mathbf{W}\mathbf{e}_{n_j}), \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a trainable matrix to perform transformation, $\mathcal{N}_{n_i}$ denotes the set of neighbors of node n_i and is adopted for normalization, and $\mathbf{e}_{n_j}$ is the embedding of node n_j collected from the embedding matrix $\mathbf{E}$.

3.5.2 Neighbor Aggregation. Next, we update the representation of a node by aggregating the propagated embeddings of its neighbors. Specifically, we utilize the sum aggregation and a non-linear activation function. Formally, let $\mathbf{e}_u$ and $\mathbf{e}_i$ denote the representations for user u and recipe i. The updating rule can be formulated as follows:

$$\begin{aligned} \hat{\mathbf{e}}_i &= \tanh\left(\mathbf{e}_i + \sum_{o \in \mathcal{N}_i} \mathbf{m}_{o \to i}\right), \\ \hat{\mathbf{e}}_u &= \tanh\left(\mathbf{e}_u + \sum_{o \in \mathcal{N}_u} \mathbf{m}_{o \to u}\right), \end{aligned} \tag{6}$$

where $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$ are the updated embeddings of user node u and recipe node i, $\mathcal{N}_i$ denotes the set of nodes that consists of the ingredient nodes of the recipe i and its calorie node, and $\mathcal{N}_u$ denotes the set of recipes that the user u has interacted with. Here, only the sum aggregation is applied, leaving the exploration of other aggregators, such as attention networks, in future work. Since a user's calorie awareness can be reflected by his/her interacted recipes, the recipe nodes can collect calorie information from the calorie nodes through embedding propagation and aggregation, where the recipes work as the bridge between users and calories.

3.6 User-guided Comprehensive Recipe Representation Learning

So far, we have obtained the calorie-aware user representation $\hat{\mathbf{e}}_u$. To predict the probability that the user u will choose the recipe i and accomplish the food recommendation task, we also need to learn the comprehensive recipe representation. Although the recipe embedding $\hat{\mathbf{e}}_i$ has already aggregated information from ingredients and calories through the graph neural networks, the user's dynamic preference of taste and calories is not considered in the recipe representation. Therefore,

following Reference [13], we learn a comprehensive user-guided recipe representation through attention networks based on both the calorie-aware user representation and the multi-modal information of the recipe.

3.6.1 Personalized Ingredient Representation. Considering that there are different numbers of ingredients in different recipes, we need to aggregate the ingredient embeddings into a single compact representation. Given that users have their personalized preferences over different ingredients and some ingredients contribute more to the taste of the recipe than other ingredients, we employ a user-guided attention to obtain the compressed ingredient representation.

Specifically, given the calorie-aware user representation $\hat{\mathbf{e}}_u$, the imaged feature $\mathbf{v}_i$ of the recipe extracted by pretrained networks and the embedding of the target ingredient $\mathbf{e}_{g_i^k}$, we first adopt a mapping layer to transform the image feature into a embedding with dimension d, which is formulated as:

$$\mathbf{p}_i = \mathbf{W}_p \mathbf{v}_i + \mathbf{b}_p. \tag{7}$$

Here, the image embedding is denoted as $\mathbf{p}_i \in \mathbb{R}^d$, $\mathbf{W}_p \in \mathbb{R}^{d \times 2048}$ and $\mathbf{b}_p$ are the parameters of the mapping layer. Then, we adopt a feed forward network to compute the attention weights of ingredients:

$$\begin{aligned} a_i^k(u) &= \mathbf{h}_1^T \tanh(\mathbf{W}_{1u}\hat{\mathbf{e}}_u + \mathbf{W}_{1v}\mathbf{p}_i + \mathbf{W}_{1g}\mathbf{e}_{g_i^k} + \mathbf{b}_1), \\ \alpha_i^k(u) &= \frac{\exp(a_i^k(u))}{\sum_k \exp(a_i^k(u))}, \text{ where } g_i^k = 1, \end{aligned} \tag{8}$$

where $W_{1*} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_1 \in \mathbb{R}^d$, and $\mathbf{h}_1 \in \mathbb{R}^d$ are the parameters to be learned. We adopt tanh as the non-linear activation function. Finally, the embeddings of the ingredients can be fused by the attention weights and we can obtain the personalized ingredient representation $\hat{\mathbf{e}}_{\mathbf{g}_i}$:

$$\hat{\mathbf{e}}_{\mathbf{g}_i} = \sum_k \alpha_i^k(u)\mathbf{e}_{g_i^k}, \text{ where } g_i^k = 1. \tag{9}$$

3.6.2 User-guided Recipe Representation. After considering the user's preference over different ingredients and obtaining the fused ingredient representation $\hat{\mathbf{e}}_{\mathbf{g}_i}$, we now have representations $(\hat{\mathbf{e}}_i, \hat{\mathbf{e}}_{\mathbf{g}_i}, \mathbf{p}_i, \mathbf{e}_c)$ corresponding to different components of the recipes. In the meantime, considering that a user may choose a recipe for its appearance, the taste, or the calories of the recipe, similar to personalized ingredient attention, we employ a component-level attention to aggregate representations of different components into a comprehensive recipe representation. Specifically, we adopt a feed forward network to compute the attention weights of different components:

$$\begin{aligned} b_{\mathbf{q}}(u) &= \mathbf{h}_2^T \tanh(\mathbf{W}_{2u}\hat{\mathbf{e}}_u + \mathbf{W}_{2\mathbf{q}}\mathbf{q} + \mathbf{b}_2), \\ \beta_{\mathbf{q}}(u) &= \frac{\exp(b_{\mathbf{q}}(u))}{\sum_{\mathbf{q} \in \{\hat{\mathbf{e}}_i, \hat{\mathbf{e}}_{\mathbf{g}_i}, \mathbf{p}_i, \mathbf{e}_{c_i}\}} \exp(b_{\mathbf{q}}(u))}, \end{aligned} \tag{10}$$

where $\mathbf{W}_{2*} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_2 \in \mathbb{R}^d$, and $\mathbf{h}_2 \in \mathbb{R}^d$ are the parameters to be learned. Note that we use $\mathbf{q}$ to represent one of the component representations to simplify the illustration.

Then, different component representations can be fused by the attention weights and we obtain the user-guided comprehensive recipe representation $\tilde{\mathbf{e}}_i$:

$$\tilde{\mathbf{e}}_i = \sum_{\mathbf{q} \in \{\hat{\mathbf{e}}_i, \hat{\mathbf{e}}_{\mathbf{g}_i}, \mathbf{p}_i, \mathbf{e}_{c_i}\}} \beta_{\mathbf{q}}(u) \cdot \mathbf{q}. \tag{11}$$

3.7 Personalized Food Recommendation

Given the user representation $\hat{\mathbf{e}}_u$ and the recipe representation $\tilde{\mathbf{e}}_i$, following Reference [24], we concatenate them with their element-wise product ($\hat{\mathbf{e}}_u \odot \tilde{\mathbf{e}}_i$). To predict how likely the user u would interact with recipe i, a **multi-layer perceptron (MLP)** is adopted to process the concatenated representation. Formally, the food recommendation prediction output can be formulated as:

$$\hat{y}_{ui} = \mathbf{h}_3^T f\left(\mathbf{W}_3 \begin{bmatrix} \hat{\mathbf{e}}_u \\ \tilde{\mathbf{e}}_i \\ \hat{\mathbf{e}}_u \odot \tilde{\mathbf{e}}_i \end{bmatrix} + \mathbf{b}_3\right), \tag{12}$$

where $\mathbf{W}_3 \in \mathbb{R}^{d \times 3d}$, $\mathbf{b}_3 \in \mathbb{R}^d$, and $\mathbf{h}_3 \in \mathbb{R}^d$ denote the learnable parameters of the output layer, and $f(\cdot)$ denotes the non-linear activation function, i.e., ReLU.

3.8 Model Training

The proposed SCHGN can be optimized in an end-to-end manner with the overall objective function as follows:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \alpha_{\text{multi}} \cdot \mathcal{L}_{\text{ssip}}, \tag{13}$$

where α_{multi} is a balance coefficient, $\mathcal{L}_{\text{ssip}}$ and $\mathcal{L}_{\text{rec}}$ are loss functions for the self-supervised masked ingredient prediction and the food recommendation, respectively, which will be introduced with more details as follows.

3.8.1 Self-supervised Ingredient Prediction Loss. Given the predicted context representation $\mathbf{H}_t$ obtained from Section 3.4 and masked ingredients $\hat{\mathbf{g}}_i$, we define the self-supervised learning loss for masked ingredient prediction as follows:

$$\mathcal{L}_{\text{ssip}} = \sum_{m=1}^{M} -\ln \sigma\left(f(\mathbf{h}_m, g_m) - f(\mathbf{h}_m, \bar{g}_m)\right), \tag{14}$$

where M denotes the number of masked ingredients, g_m is a masked ingredient in $\hat{\mathbf{g}}_i$, $\mathbf{h}_m \in \mathbb{R}^d$ denotes the corresponding predicted embedding of g_m collected from $\mathbf{H}_t$, $\bar{g}_m$ denotes the ingredient randomly sampled from the overall ingredient set excluding the masked ingredients, and σ represents the Sigmoid function. We implement the $f(\cdot, \cdot)$ as follows:

$$f(\mathbf{h}_m, g_m) = \mathbf{h}_m \mathbf{W}_m \mathbf{e}_{g_m}^T, \tag{15}$$

where $\mathbf{W}_m \in \mathbb{R}^{d \times d}$ denotes parameters of the linear transform, and $\mathbf{e}_{g_m} \in \mathbb{R}^d$ is the original representation of g_m collected from $\mathbf{X}_{\mathbf{g}_i}$.

3.8.2 Recommendation Loss. To learn user's preferences on different recipes, we adopt **Bayesian Personalized Ranking (BPR)** as our loss function for food recommendation, which has been widely used in recommendation tasks [23, 47]. BPR assumes that the observed interaction has higher prediction scores than unobserved ones. The objective function can be formulated as:

$$\mathcal{L}_{\text{rec}} = \sum_{(u,i,j) \in O} -\ln(\sigma(\hat{y}_{ui} - \hat{y}_{uj})) + \lambda \|\theta\|^2, \tag{16}$$

where σ represents the Sigmoid function and λ is a hyper-parameter for model regularization, θ denotes all learnable parameters in our model, and O denotes the set of positive-negative sample pairs that consists of triples in the form (u, i, j), where u denotes the user together with an interacted recipe i and a non-observed recipe j.

Table 1. Example of the Calorie Information in Allrecipes Dataset

Recipe	Calorie Information
Potato Bacon Pizza	"calories": {"hasCompleteData": True, "name": "Calories," "amount": 162.6685, "percentDailyValue": 8, "displayValue": 163, "unit": "kcal"}

4 EXPERIMENTS

In this section, we first introduce the dataset and experimental settings. We then report the performance of our proposed method compared with baselines and adopt an ablation study to investigate the effectiveness of the calorie factor and the self-supervised ingredient prediction. At last, we conduct experiments to compare different methods on recipes of different popularity levels and explore the impact of hyper-parameters of our method. We make the source code publicly available.[3]

4.1 Dataset

To demonstrate the effectiveness of our proposed method, we conduct experiments on a real-world dataset for food recommendation. In this article, the input of our food recommendation model includes users, recipes, ingredients, recipe images, and calories of the recipes. To train the recommendation model, we also need the user-recipe interactions for the training data. Allrecipes,[4] built by Gao et al. [13], is the only available large-scale dataset that meets our requirements, which is crawled from a recipe-sharing platform Allrecipes.com. There are 68,768 users, 45,630 recipes with 33,147 ingredients, and 1,093,845 interactions. For each recipe in Allrecipes, there is a corresponding nutritional fact column, which provides the calorie information. We provide an example of calorie information in Table 1. Since Allrecipes does not explain how to calculate the calorie information of each recipe, we are not sure whether there is missing nutrition information of ingredients. However, each recipe in the dataset has its corresponding calorie information provided by the Allrecipes, and we only use the recipe-level calorie information in our experiment. We follow the data partition defined in Reference [13], where the test set includes the latest 30% of interaction history and the remaining data are split into training (60%) and validation (10%) sets.

4.2 Experimental Settings

4.2.1 Evaluation Metrics. We employ three popular evaluation metrics to evaluate the performance of food recommendation including **Area Under the Roc Curve (AUC), Normalized Discounted Cumulative Gain (NDCG)**, and Recall. Given a user u and a pair of positive-negative recipes (i, j), AUC measures the probability that a recommender will rank a positive recipe i higher than a negative j. NDCG@k is a widely used measure to evaluate the quality of the ranked list. Recall@K measures the proportion that positive recipes are ranked in top-K recommended recipes. Since the recipe set is too large, it is time-consuming to rank all recipes when testing. Following the evaluation strategy adopted in Reference [13], there are 500 sampled negative recipe instances for one user and the interacted recipes in the test set. To ensure the robustness of the experimental results, we repeat each evaluation 10 times and report the mean value as the final performance.

4.2.2 Implementation Details. Given the raw images of recipes, we extract the image features by ResNet-50 [21] and use the output of *pool5* layer as the input image feature, which is a 2,048 dimension vector. The ResNet-50 is pre-trained on ImageNet [7] and fine-tuned via classifying raw

[3] https://github.com/TAEYOUNG-SYG/SCHGN.
[4] https://www.kaggle.com/elisaxxygao/foodrecsysv1.

recipe images into their associated categories (e.g., chicken rice) as in Reference [13]. We select the optimal hyper-parameters of our model according to the metric of AUC on the validation set. Dimension of the embedding for users, recipes, ingredients, and calories is 64. For the self-supervised masked ingredient prediction, we set the probability of masking as 0.2 and the weight (i.e., the balance coefficient in Equation (13)) for the self-supervised loss as 0.008. The number of the multi-head self-attention, feed forward layers, and attention heads are set to 2. For the calories of recipes, we transform the continuous value of calorie into separate levels using uniform quantization. Specifically, the calories of recipes range from 50 to 4,700 and we divide them into 10 calorie levels. For example, for a certain recipe with the calories of 1,000, its calorie level is $\lfloor \frac{1,000-50}{4,700-50} \times 10 \rfloor = 2$. For the training of our model, we use Adam optimizer [29] with a learning rate of 0.0005 and train the model for 30 epochs with the batch size of 512. The weight decay is set to 0.01, 0.1, 0.5, and 0.01 for embeddings (users, recipes, ingredients, and calories), weights of the image mapping layer, weights of the MLP, and weights of the GNNs, respectively.

4.2.3 Comparison Methods. To verify the effectiveness of our proposed method, we compare the performance with the following baselines:

- LDA [56]: This method adopts a classical probabilistic factorization model, **Latent Dirichlet Allocation (LDA)**, where users are regarded as documents and recipes as words.
- MF-BPR [47]: This is a standard matrix factorization method optimized by **Bayesian Personalized Ranking (BPR)** loss. It uses the ID embeddings for recipes without considering content information.
- FM [46]: The **Factorization Machine (FM)** is a competitive model that applies a sum of pairwise inner products of user and item features to obtain the prediction score.
- VBPR [22]: This method considers the visual feature of the recipe item and incorporates the pre-trained image feature into MF-BPR.
- FM-VBPR [13]: This method incorporates visual features into factorization machines.
- PUP [73, 74]: This method is a graph-based solution for general recommendation, which incorporates the price and category of items into the user-item graph and explicitly captures the user's preference on price with GCNs. For food recommendation, we replace the price and category nodes in the graph with calorie and ingredient nodes.
- HAFR [13]: Given the user-recipe interactions, recipe ingredients, and recipe images, this method adopts a neural network solution equipped with hierarchical attention mechanism to fully investigate the impact of different factors on user's food decision-making process. Considering the definition of our task, HAFR is the only food recommendation method that meets the requirements.
- Cal-HAFR: This method incorporates the calorie factor to HAFR, where the calories are used as a component of the recipe to learn the recipe representation with hierarchical attention. We term this method as Cal-HAFR. We design this method to show whether the calorie-aware food recommendation can be simply accomplished by the state-of-the-art method.

We add a summary of different methods in Table 2. As shown, we not only compare our method with existing state-of-the-art food recommendation methods (e.g., FM-VBPR and HAFR), but also compare with popularly used recommendation methods in other domains, e.g., movies and E-commerce.

4.3 Performance Comparison

We first compare the proposed method with all the baselines on Allrecipes dataset with respect to the metric of AUC, NDCG@10, and Recall@10. Table 3 reports the results on personalized food

Table 2. Summary of the Compared Methods in Our Experiment

Method	Recipe Image	Ingredient	Calorie	Method Type	Domain
LDA [56]	×	×	×	LDA	Text
MF [47]-BPR	×	×	×	MF	Movies
FM [46]	×	√	×	FM	Movies
VBPR [22]	√	×	×	MF	E-commerce
FM-VBPR [13]	√	√	×	FM	Food
PUP [73, 74]	√	√	√	GNN	E-commerce
HAFR [13]	√	√	×	NN	Food
Cal-HAFR	√	√	√	NN	Food
Ours	√	√	√	GNN	Food

"LDA" denotes "Latent Dirichlet Allocation." "MF" denotes "Matrix Factorization." "FM" denotes "Factorization Machine." "GNN" denotes "Graph Neural Network." "NN" denotes "Neural Network." "Domain" represents the field in which the method was first used.

Table 3. Performances of Different Methods for Personalized Food Recommendation on the Allrecipes Dataset

Methods	Allrecipes		
	AUC	NDCG@10	Recall@10
LDA	0.5154	0.0376	0.0601
MF-BPR	0.5622	0.0376	0.0567
FM	0.5710	0.0396	0.0607
VBPR	0.5808	0.0296	0.0431
FM-VBPR	0.5840	0.0372	0.0580
PUP	0.6526	0.0441	0.0676
HAFR	0.6435	0.0455	0.0674
Cal-HAFR	0.6562	0.0482	0.0708
Ours	**0.7212***	**0.0569***	**0.0883***
Improvement%	9.91%	18.05%	24.72%

Higher is better for all metrics. *denotes the statistical significance for $p < 0.05$.

recommendation. It is worth noting that we show the experimental results of LDA, MF-BPR, FM, VBPR, FM-VBPR, HAFR reported in Reference [13]. To compare with PUP [73, 74], we reproduce the method based on the article and report the performance on Allrecipes dataset. From the results, we have the following observations:

- MF and LDA perform worse than other baselines. Both of the two methods predict the user preference on one recipe only based on user-recipe interactions, which justifies the benefit of modeling the recipe content.
- Among the baselines, Cal-HAFR performs the best, which verifies the advantage of jointly modeling the user-recipe interactions and the recipe contents. Besides, compared with

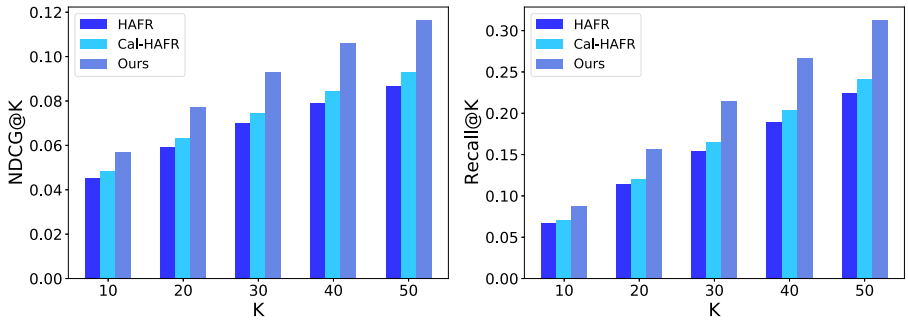

Fig. 3. Performances of HAFR, Cal-HAFR, and our method in Top-K recipe recommendation with different settings of K on NDCG@K and Recall@K metrics.

Table 4. Ablation Study on the Importance of the Self-supervised Ingredient Prediction and the Modeling of the Calorie Factor

Method	AUC	NDCG@10	Recall@10	NDCG@20	Recall@20	NDCG@50	Recall@50
Ours w/o ssip	0.6982	0.0464	0.0710	0.0643	0.1312	0.1004	0.2752
Ours w/o cal	0.6860	0.0472	0.0714	0.0658	0.1339	0.0995	0.2692
Ours	**0.7212**	**0.0569**	**0.0883**	**0.0774**	**0.1570**	**0.1165**	**0.3129**

Ours w/o cal represents the method without considering calorie information. Ours w/o ssip represents the method without using the self-supervised ingredient prediction.

HAFR, Cal-HAFR achieves a superior performance, which demonstrates the effectiveness of incorporating calorie factor of the recipe in food recommendation.

- By comparing our method with all the baselines, it is clear to see that Ours performs consistently better than others by a large margin over all metrics. Specifically, Ours achieves relative improvements of 9.91%, 18.05%, 24.72% over the strongest baseline (i.e., Cal-HAFR), regarding to AUC, NDCG@10, and Recall@10. The results justify the effectiveness of our framework, where we build a heterogeneous graph to explicitly capture users' calorie awareness and model the complex relationships between ingredients with self-supervised learning.

As illustrated in Figure 3, we present the performance of Top-K recipe recommendation where the ranking position K ranges from 10 to 50. It can be seen that our method consistently performs better than HAFR and Cal-HAFR on both NDCG and Recall metrics. The results demonstrate the robustness of our method.

4.4 Ablation Study

To better understand the proposed method, we further evaluate its key contributions, i.e., the self-supervised ingredient prediction module and the incorporated calorie factor. Specifically, we compare two variants of Ours by removing the self-supervised ingredient prediction (Ours w/o ssip) and the calorie information of the recipe (Ours w/o cal). We present the performance of Ours and its variants in Table 4. From the results, we have the following observations:

- By applying the self-supervised ingredient prediction, the full method outperforms Ours w/o ssip. The good performance verifies that adopting the masked ingredient prediction is

Table 5. Ablation Study on Calorie Information of Different User Groups

User Groups	Method	AUC	NDCG@10	Recall@10
A	Ours	0.6925	0.0462	0.0713
	Ours w/o cal	0.6473	0.0390	0.0607
B	Ours	0.6615	0.0406	0.0393
	Ours w/o cal	0.6589	0.0418	0.0383

Users with biased calorie preference belong to group A. Users with unbiased calorie preference belong to group B.

beneficial for the food recommendation. It also indicates the effectiveness of modeling the relationship between ingredients, which further enhances recipe representations.

- Compared with Ours w/o cal, the full method achieves a superior performance. The good performance demonstrates that it is necessary and crucial to incorporate the calorie factor of recipes into the food recommendation.

Besides, we further conduct an experiment based on different user groups to directly demonstrate the effectiveness of the proposed method in considering the user's preference on calories. We first divide users into two groups, those with biased calorie preference (Group A) and those with unbiased calorie preference (Group B). The division standard is that, for a user, if the difference between the maximum and minimum calorie level of the foods he has chosen is less than or equal to a fixed threshold (set to 1 in the experiment), the user belongs to the group with biased calorie preference. Otherwise, the user belongs to the other group. Then, we build the training, validation, and test datasets for each user group. We independently conduct the ablation study on different user groups. We report the results in Table 5. We can see that if we remove the calorie data from a group of users with biased calorie preference (A), then the performance of our model decreases a lot. For the group of users with unbiased calorie preference (B), considering calorie information has relatively less impact on the performance. The experimental results demonstrate the effectiveness of the proposed method. In addition, although the food calorie depends on the content and amounts of the ingredients, the results demonstrate that it is not easy to directly infer the calorie information from the ingredients to help the recommendation model without explicit calorie annotations.

4.5 Performances over Recipes of Different Popularity Levels

As we all know, there is a big difference in the popularity of different recipes, which will significantly affect the recommendation performance of the target recipe. To investigate the robustness of different methods on this problem, we evaluate the performance over recipes with different popularity levels. Specifically, we divide the test set into four groups based on the number of ratings of each recipe, which is denoted as N. Then, we evaluate different methods over the four groups under the metric of AUC. The performance of different methods is presented in Figure 4. We have the following observations:

- The proposed method outperforms two strong baselines (HAFR and Cal-HAFR) in all the groups consistently, which demonstrates the effectiveness and robustness of our method. Modeling the relations between ingredients leads to better recipe representations. Besides, Cal-HAFR achieves better results than HAFR, which again verifies the necessity of considering users' calorie preference for food recommendation.

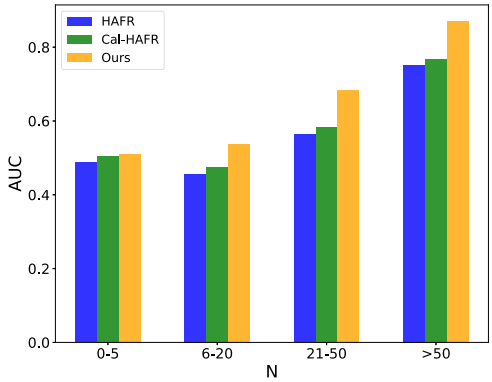

Fig. 4. Performances over recipes with different popularity levels. N denotes the number of ratings of each recipe.

- In general, the performance of models increases with more ratings for each recipe, especially for the group with $N > 50$, where all methods have significant improvements. The results show that users' preference is more easy to be captured with more historical user-recipe interactions. The improvement of our method is more obvious, which again shows the superiority of our method in user preference modeling with sufficient data.

4.6 Evaluation of Calorie Preference Modeling

We add an evaluation metric based on ground-truth calories, which is named as **Calorie Error Ratio (CER)**. The CER can be computed as follows:

$$\text{CER} = \frac{1}{N_U} \sum_{u}^{N_U} \frac{|\text{Cal}^u_{\text{pred}} - \text{Cal}^u_{\text{gt}}|}{\text{Cal}^u_{\text{gt}}}, \tag{17}$$

where $\text{Cal}^u_{\text{pred}}$ denotes the calorie of the recipe with the highest recommendation score, Cal^u_{gt} denotes the ground-truth calorie, and N_U denotes the number of users.

Table 6 reports the CER results of our model and the existing state-of-the-art food recommendation method HAFR. As shown, our model achieves better results than HAFR, which demonstrates the effectiveness our model in modeling the users' preference of calories.

4.7 Hyper-parameter Study

We study the effect of two important hyper-parameters on the performance of our proposed method.

4.7.1 Fineness of the Calorie Factor. Calories of the recipe collected from food-sharing platforms are generally continuous. In our work, we discretize the calorie information into separate levels using uniform quantization, as illustrated in Section 4.2.2. The number of calorie levels is a crucial hyper-parameter, which decides the fineness of the calorie factor in our proposed method. We conduct experiments on different calorie levels to study how the granularity of the calorie factor influences the food recommendation performance. We present the results with different calorie levels in Figure 5. The number of calorie levels ranges from 2 to 100, and we adopt AUC, NDCG@10, and Recall@10 as the metrics. It can be seen that the proposed method achieves the best performance when the number of calorie level is set as 10. When we set the number of calorie levels as 2 or 5, the calorie factor is not properly incorporated, hence the overall performance is

Table 6. Comparison of Ours and HAFR on CER

Method	CER
HAFR	0.7824
Ours	0.7206

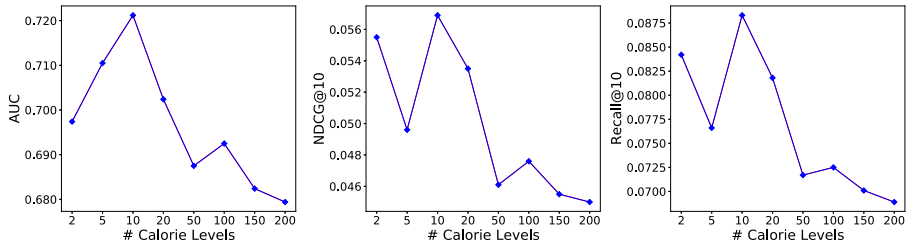

Fig. 5. Performances of our method with different number of calorie levels.

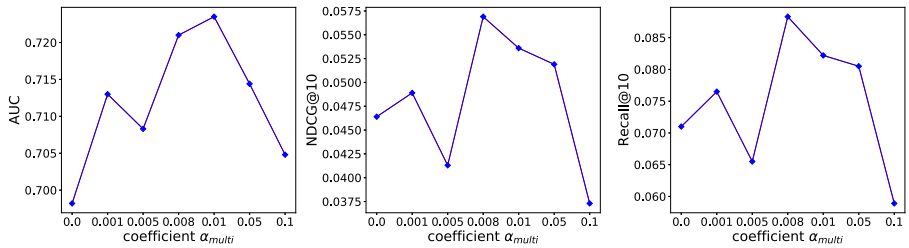

Fig. 6. Performances of our method with different settings of the balance coefficient α_{multi} defined in the objective function.

inferior to the best model. However, if we raise the number of calorie levels too high, then the overall trend is declining, because too many calorie levels will lead to long-tailed classes that are difficult to correctly recognize. The model with 50 levels performs slightly worse than that with 100 because some calorie levels may have too large intra-class variation.

4.7.2 Effect of the Coefficient α_{multi} in Equation (13). The coefficient α_{multi} in Equation (13) balances the importance of the self-supervised ingredient prediction loss and the recommendation loss. To explore the impact, we investigate the model performance by adjusting the coefficient α_{multi} during training. We present the performance of our method with different values of the coefficient α_{multi} in Figure 6. It can be seen that our model achieves the best performance when we set the coefficient as 0.008. When the coefficient has smaller value, the model's performance decreases, since the self-supervised ingredient prediction is not properly trained. When we increase the coefficient larger than the optimal setting, the performance also decreases, since focusing too much on self-supervised ingredient prediction will definitely hurt the final recommendation.

4.8 Time Efficiency Analysis

We provide the time efficiency of our method and Cal-HAFR, which performs the best among the baselines. As shown in Table 7, it takes more time for our model in the training phase due

Table 7. Comparison of Ours and Cal-HAFR on Time Efficiency

Method	Train	Inference
Cal-HAFR	0.03 sec/batch	0.01 sec/user
Ours	0.10 sec/batch	0.01 sec/user

to the graph structure and self-supervised ingredient prediction. During the inference, our model achieves better results while maintaining the same time cost as the baseline method.

5 CONCLUSION

In this article, we propose a **Self-supervised Calorie-aware Heterogeneous Graph Network (SCHGN)** to solve the task of food recommendation, which can better model the relations of ingredients and capture the user's preference on food calories simultaneously. To be specific, we build a heterogeneous graph to explicitly present the complex relations among users, recipes, ingredients, and calories. We explore the co-occurrence relation of ingredients in different recipes via self-supervised ingredient prediction. Meanwhile, we highlight the significance of the user's preference on calories in food decision-making process and learn calorie-aware user representations with hierarchical message passing to dynamically explore user's preference of taste and calories. Extensive experiments have been conducted by comparing our method with other competing methods on a benchmark dataset Allrecipes. The experimental results demonstrate that our method consistently outperforms the state-of-the-art models. Besides, ablation experiments verify the usefulness of the proposed self-supervised ingredient prediction and the incorporated calorie factor. In the future, we will (1) investigate how to make full use of multimodal information of recipe to learn better recipe representations with self-supervised learning; (2) incorporate the side information of users (e.g., age and health) and other nutrition factors of food into the recommendation framework.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17, 6 (2005), 734–749.

[2] Ahmed Al-Nazer, Tarek Helmy, and Mohammed Al-Mulhem. 2014. User's profile ontology-based semantic framework for personalized food and nutrition recommendation. *Procedia Comput. Sci.* 32 (2014), 101–108.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*. PMLR, 1597–1607.

[5] Yu Chen, Ananya Subburathinam, Ching-Hua Chen, and Mohammed J. Zaki. 2021. Personalized food recommendation as constrained question answering over a large-scale food knowledge graph. In *14th ACM International Conference on Web Search and Data Mining*. 544–552.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375* (2016).

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] David Elsweiler, Morgan Harvey, Bernd Ludwig, and Alan Said. 2015. Bringing the "healthy" into food recommenders. In *DMRS*. 33–36.

[10] David Elsweiler, Christoph Trattner, and Morgan Harvey. 2017. Exploiting food choice biases for healthier recipe recommendation. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 575–584.

[11] Jill Freyne and Shlomo Berkovsky. 2010. Intelligent food planning: Personalized recipe recommendation. In *15th International Conference on Intelligent User Interfaces*. 321–324.
[12] Jill Freyne and Shlomo Berkovsky. 2010. Recommending food: Reasoning on recipes and ingredients. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 381–386.
[13] Xiaoyan Gao, Fuli Feng, Xiangnan He, Heyan Huang, Xinyu Guan, Chong Feng, Zhaoyan Ming, and Tat-Seng Chua. 2019. Hierarchical attention network for visually aware food recommendation. *IEEE Trans. Multim.* 22, 6 (2019), 1647–1659.
[14] Mouzhi Ge, Mehdi Elahi, Ignacio Fernaández-Tobías, Francesco Ricci, and David Massimo. 2015. Using tags and latent factors in a food recommender system. In *5th International Conference on Digital Health*. 105–112.
[15] Mouzhi Ge, Francesco Ricci, and David Massimo. 2015. Health-aware food recommender system. In *9th ACM Conference on Recommender Systems*. 333–334.
[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).
[17] Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R. Varshney. 2020. RecipeGPT: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proceedings of the Web Conference*. 181–184.
[18] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
[19] Morgan Harvey, Bernd Ludwig, and David Elsweiler. 2013. You are what you eat: Learning user tastes for rating prediction. In *International Symposium on String Processing and Information Retrieval*. Springer, 153–164.
[20] Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L. McGuinness, and Mohammed J. Zaki. 2019. FoodKG: A semantics-driven knowledge graph for food recommendation. In *International Semantic Web Conference*. Springer, 146–162.
[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[22] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian personalized ranking from implicit feedback. In *AAAI Conference on Artificial Intelligence*.
[23] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.
[24] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *26th International Conference on World Wide Web*. 173–182.
[25] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).
[26] Xiaowen Huang, Shengsheng Qian, Quan Fang, Jitao Sang, and Changsheng Xu. 2020. Meta-path augmented sequential recommendation with contextual co-attention network. *ACM Trans. Multim. Comput., Commun., Applic.* 16, 2 (2020), 1–24.
[27] Longlong Jing and Yingli Tian. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).
[28] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored item similarity models for top-n recommender systems. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 659–667.
[29] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[30] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[31] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1920–1929.
[32] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 426–434.
[33] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[34] Diya Li and Mohammed J Zaki. 2020. Reciptor: An effective pretrained model for recipe representation learning. In *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1719–1727.
[35] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical fashion graph network for personalized outfit recommendation. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 159–168.

[36] Chia-Jen Lin, Tsung-Ting Kuo, and Shou-De Lin. 2014. A content-based matrix factorization model for recipe recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 560–571.
[37] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.
[38] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 1 (2019), 187–203.
[39] Lei Meng, Fuli Feng, Xiangnan He, Xiaoyan Gao, and Tat-Seng Chua. 2020. Heterogeneous fusion of semantic and collaborative information for visually aware food recommendation. In *28th ACM International Conference on Multimedia*. 3460–3468.
[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* 26 (2013), 3111–3119.
[42] Weiqing Min, Shuqiang Jiang, and Ramesh C. Jain. 2019. Food recommendation: Framework, existing solutions and challenges. *IEEE Trans. Multim.* (2019).
[43] Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6707–6717.
[44] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*. Springer, 69–84.
[45] Erik Quintanilla, Yogesh Singh Rawat, Andrey Sakryukin, Mubarak Shah, and Mohan Kankanhalli. 2020. Adversarial learning for personalized tag recommendation. *IEEE Trans. Multimedia* (2020).
[46] Steffen Rendle. 2012. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3, 3 (2012), 1–22.
[47] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *25th Conference on Uncertainty in Artificial Intelligence*. 452–461.
[48] Amaia Salvador, Erhan Gundogdu, Loris Bazzani, and Michael Donoser. 2021. Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15475–15484.
[49] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning*. PMLR, 843–852.
[50] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-BERT: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530* (2019).
[51] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* 2009 (2009).
[52] V. Subramaniyaswamy, Gunasekaran Manogaran, R. Logesh, V. Vijayakumar, Naveen Chilamkurti, D. Malathi, and N. Senthilselvan. 2019. An ontology-driven personalized food recommendation in IoT-based healthcare system. *J. Supercomput.* 75, 6 (2019), 3184–3216.
[53] Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. 2012. Recipe recommendation using ingredient networks. In *4th Annual ACM Web Science Conference*. 298–307.
[54] Thi Ngoc Trang Tran, Müslüm Atas, Alexander Felfernig, and Martin Stettinger. 2018. An overview of recommender systems in the healthy food domain. *J. Intell. Inf. Syst.* 50, 3 (2018), 501–526.
[55] Christoph Trattner and David Elsweiler. 2017. Food recommender systems: Important contributions, challenges and future research directions. *arXiv preprint arXiv:1711.02760* (2017).
[56] Christoph Trattner and David Elsweiler. 2017. Investigating the healthiness of internet-sourced recipes: Implications for meal planning and recommender systems. In *26th International Conference on World Wide Web*. 489–498.
[57] Kosetsu Tsukuda, Takehiro Yamamoto, Satoshi Nakamura, and Katsumi Tanaka. 2010. Plus one or minus one: A method to browse from an object to another object by adding or deleting an element. In *International Conference on Database and Expert Systems Applications*. Springer, 258–266.
[58] Mayumi Ueda, Syungo Asanuma, Yusuke Miyawaki, and Shinsuke Nakajima. 2014. Recipe recommendation method by considering the users preference and ingredient quantity of target recipe. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1. 12–14.
[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
[60] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[61] Wenjie Wang, Ling-Yu Duan, Hao Jiang, Peiguang Jing, Xuemeng Song, and Liqiang Nie. 2021. Market2Dish: Health-aware food recommendation. *ACM Trans. Multim. Comput., Commun., Applic.* 17, 1 (2021), 1–19.

[62] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.

[63] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[64] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. 2020. Self-supervised reinforcement learning for recommender systems. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 931–940.

[65] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[66] Ming Yan, Jitao Sang, Changsheng Xu, and M. Shamim Hossain. 2016. A unified video recommendation by cross-network user modeling. *ACM Trans. Multim. Comput., Commun., Applic.* 12, 4 (2016), 1–24.

[67] Longqi Yang, Yin Cui, Fan Zhang, John P. Pollak, Serge Belongie, and Deborah Estrin. 2015. PlateClick: Bootstrapping food preferences through an adaptive visual interface. In *24th ACM International on Conference on Information and Knowledge Management*. 183–192.

[68] Longqi Yang, Cheng-Kang Hsieh, Hongjian Yang, John P. Pollak, Nicola Dell, Serge Belongie, Curtis Cole, and Deborah Estrin. 2017. Yum-me: A personalized nutrient-based meal recommender system. *ACM Trans. Inf. Syst.* 36, 1 (2017), 1–31.

[69] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[70] Satoshi Yokoi, Keisuke Doman, Takatsugu Hirayama, Ichiro Ide, Daisuke Deguchi, and Hiroshi Murase. 2015. Typicality analysis of the combination of ingredients in a cooking recipe for assisting the arrangement of ingredients. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.

[71] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* 52, 1 (2019), 1–38.

[72] Zhou Zhao, Qifan Yang, Hanqing Lu, Tim Weninger, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Social-aware movie recommendation via multimodal network learning. *IEEE Trans. Multim.* 20, 2 (2017), 430–440.

[73] Yu Zheng, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2021. Incorporating price into recommendation with graph convolutional networks. *IEEE Trans. Knowl. Data Eng.* (2021).

[74] Yu Zheng, Chen Gao, Xiangnan He, Yong Li, and Depeng Jin. 2020. Price-aware recommendation with graph convolutional networks. In *IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 133–144.

[75] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *29th ACM International Conference on Information & Knowledge Management*. 1893–1902.

Received 11 September 2021; revised 21 January 2022; accepted 8 March 2022