

Heterogeneous Fusion of Semantic and Collaborative Information for Visually-Aware Food Recommendation

Lei Meng

¹Shandong University

²National University of Singapore
menglei.thunder@gmail.com

Fuli Feng

National University of Singapore
dcsfeng@nus.edu.sg

Xiangnan He

University of Science and Technology
of China
xiangnanhe@gmail.com

Xiaoyan Gao

Beijing Institute of Technology
xygao@bit.edu.cn

Tat-Seng Chua

National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Users' selection of food has been verified to be vision-driven. This leads to visually-aware food recommendation, which recommends food items based on their visual features. Existing methods typically use the pre-extracted visual features from food classification models, which mainly encode the visual content with limited semantic information, such as the classes and ingredients. Therefore, such features may not cover the personalized visual preferences of users, termed collaborative information, e.g. users may attend to different colors and textures of food based on their preferred ingredients and cooking methods. To address this problem, this paper presents a heterogeneous multi-task learning framework, termed privileged-channel infused network (PiNet). It learns the visual features that contain both the semantic and collaborative information by training the image encoder to simultaneously fulfill the ingredient prediction and food recommendation tasks. However, the heterogeneity between the two tasks may lead to different visual information in need and different directions in model parameter optimization. To handle these challenges, PiNet first employs a dual-gating module (DGM) to enable the encoding and passing of different visual information from the image encoder to individual tasks. Secondly, PiNet adopts a two-phase training strategy and two prior knowledge incorporation methods to ensure an effective model training. Experimental results from two real-world datasets show that the visual features generated by PiNet better attend to the informative image regions, yielding superior performance.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Visually-aware recommendation; Personalized visual preference; Heterogeneous multi-task learning; Dual-gating module

ACM Reference Format:

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 *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3394171.3413598>

1 INTRODUCTION

Visual food logging [15, 23, 26] as an emerging application for diet management enables users to upload photos of their daily food intake. Beyond pure diet logging, different machine learning techniques have been developed for food content analysis and personalized service provision [13, 17, 21–23, 37–39]. Food recommendation [7, 9, 29, 37] is one of the key services, which models users' eating preferences from their interactions with food items and subsequently recommends similar ones. Past efforts are usually based on the explicit feedback such as user ratings [8, 11, 18, 29] and the recipe content such as the ingredients [10, 16, 18, 28, 30, 31, 34]. However, in the visual food logging systems, users may not provide recipe information for their uploaded images, and the typical user interactions are implicit feedback, such as likes and comments.

These issues motivated the visually-aware food recommendation [7, 9, 36, 37], which uses food images and users' implicit feedback for recommendation. Recent studies [7, 36] have revealed that image is an important modality in food recommendation since users' selection of food is typically vision-driven. Existing methods typically extend the conventional collaborative filtering algorithms to learn the item embedding from the pre-extracted visual features, which are from food classification models. Therefore, these features encode mainly the visual information related to food classes or ingredients. However, users' choices of food may not be literally based on such semantic information, but also how the food looks [7, 36]. Therefore, such visual embeddings may not attend to the visual content that captures users' personalized visual preferences. As shown in Figure 1, a classifier trained for ingredient prediction extracts the features mainly from ingredient-intensive regions, while that for recommendation attends to the common visual elements of the images, namely, the white and red content, referred to as collaborative information [2]. As such, it is necessary to learn an image encoder that can preserve both types of information.

To this end, this paper presents a heterogeneous multi-task learning framework, termed privileged-channel infused network (PiNet), which learns such an image encoder by making it fulfill both the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '20, October 12–16, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7988-5/20/10...\$15.00

<https://doi.org/10.1145/3394171.3413598>



Figure 1: Semantic information v.s. collaborative information. The ingredient prediction and food recommendation tasks may attend to very different visual information.

ingredient prediction and food recommendation tasks simultaneously. Notably, simply combining the models for these two tasks into a joint one may lead to failure in both tasks, since the heterogeneous tasks may require the visual information from different image regions as illustrated in Figure 1. This makes the image encoder difficult to preserve all the visual information required by them. To address this issue, PiNet employs a dual-gating module (DGM) to control the information passing between the image encoder and the two tasks. Specifically, in the forward pass, DGM uses two task gates to reshape the unified image embedding produced by the image encoder to learn the task-aware item embeddings. In the backward propagation, DGM utilizes a gradient gate to fuse the gradients passing from both tasks to optimize the image encoder. This handles the trade-off for the image encoder between fusing the ingredient and collaborative information into a single visual embedding. Moreover, the image encoder may be optimized in different directions by heterogeneous tasks, making the model difficult to converge in the early training stage and thus leading to the degraded performance in both tasks. PiNet therefore employs a two-phase training strategy and two prior knowledge incorporation methods to ensure an effective model training. To summarize, this paper includes three main contributions:

- (1) It proposes a heterogeneous multi-task learning framework, i.e. PiNet, to learn the visual features of food images that can fuse both the semantic and collaborative information for improved recommendation performance.
- (2) A dual-gating module is proposed to enable the joint training of the heterogeneous tasks of ingredient prediction and food recommendation. It is a general method and may be extended for any heterogeneous multi-task learning problems.
- (3) Considering that only a western food dataset has been published for the visually-aware food recommendation, a new dataset on Chinese food is created to complement to the community.

2 RELATED WORK

Food Recommendation. Food recommendation aims to recommend to a user the food items that match their preferences. It is usually achieved by analyzing a user’s interactions with food items. User rating is an explicit feedback of user preference. Existing methods following this line of research [8, 11, 18, 29] form the user-item interactions with positive/negative ratings. They usually use collaborative filtering algorithms for recommendation.

Recipes have been extensively investigated in the literature for food recommendation. The rich information therein, such as the ingredients, cooking methods, and nutrition composition, describes food items at the semantic level and makes a direct link to users’ preferences. Most of the existing methods [10, 16, 18, 28, 30, 31, 34] usually use the ingredients to measure the similarity between recipes, and then recommend users with similar ones. Interestingly, nutrition composition has gained much attention as a measure to either penalize or filter out the unhealthy food [7, 11, 29, 37].

Food images usually reveal important information of food, such as the ingredients and cooking methods. Existing algorithms [7, 9, 36, 37] typically use image features as item embedding and then employ either search or collaborative filtering methods for food recommendation. Notably, all of them use pre-extracted visual features. As illustrated in Figure 1, these features may not cover the visual elements that matching the users’ personalized visual preferences.

Visually-Aware Recommendation. Visually-aware recommendation refers to the recommendation tasks delving into the visual characteristics of items. Besides the applications to food domain as discussed above, existing studies also investigate the recommendation of fashion clothes [4, 12, 14, 24, 40], E-commerce products [4, 5], restaurants [6], and point-of-interests (POIs) [35]. Despite their use of different recommendation models, most of them use the pre-extracted visual features for their respective downstream tasks. Only one study [14] explores learning the visual features for recommendation in an end-to-end manner.

3 PROBLEM STATEMENT

This study aims at encoding the semantic and collaborative visual features of food images to capture users’ personalized visual preferences. It is motivated by the following observations:

- (1) **Conventional recommenders model the item embeddings for collaborative similarity:** Given the sets of users \mathcal{U} , items \mathcal{I} , and their interaction pairs (u, i) where $u \in \mathcal{U}$ and $i \in \mathcal{I}$, a recommender $\mathcal{G}(\cdot)$ is trained to learn the latent embeddings of users and images, denoted as \mathbf{p}_u and \mathbf{q}_i , respectively. A matching score $\hat{y}_{ui} = \mathcal{G}(u, i)$ is then computed such that $\hat{y}_{ui} > \hat{y}_{uj}$ holds for $\{\exists i, j | i \in \mathcal{I}_u^+, j \in \mathcal{I} \setminus \mathcal{I}_u^+\}$ where \mathcal{I}_u^+ denotes the set of items that the user u has interacted with before, such as posts, clicks, likes, and comments. Matrix factorization-Bayesian personalized ranking (MF-BPR) [25] formulates the predictive model as $\mathcal{G}(u, i) = \mathbf{p}_u^T \mathbf{q}_i$ (note that the bias parameters are omitted for simplicity), so the higher $\mathcal{G}(u, i)$ is, the more likely u has interacted with i . Therefore, the collaborative similarity reflects the density of the shared links between users and items.
- (2) **Visually-aware recommenders typically learn the item embeddings from the pre-extracted visual features:** Existing visually-aware recommenders usually extend the conventional collaborative filtering algorithms to model the visual item embeddings from the pre-extracted visual features, where the items belonging to the same class are closer in the feature space. For example, MF-BPR achieves this by replacing the latent item embedding \mathbf{q}_i with $\zeta(\mathbf{v}_i)$ where \mathbf{v}_i denotes the pre-extracted visual features and $\zeta(\cdot)$ is a neural network-based dimensional reduction mapping. In contrast, VBPR [12] models both the latent and visual embeddings for users and images.

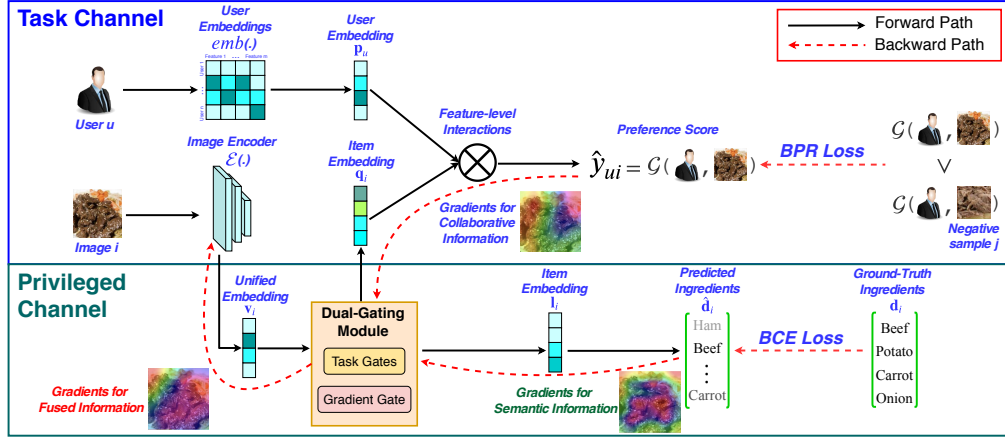


Figure 2: Illustration of PiNet that learns to fuse the semantic and collaborative information for visual feature extraction. In forward path, image encoder $\mathcal{E}(\cdot)$ and DGM extract the task-aware embeddings q_i and l_i for food recommendation and ingredient prediction, respectively. In backward path, DGM fuses the gradients from both tasks to optimize $\mathcal{E}(\cdot)$.

Based on the above discussion, a straightforward question is whether the semantic information encoded in the pre-extracted visual features contains sufficient information to learn the item embeddings for collaborative similarity? To answer this, this paper investigates the semantic and collaborative information contained in food images. As illustrated in Figure 1, the pre-extracted visual features may not be sufficient to cover the food regions of collaborative information. This motivates this study to propose the PiNet framework, which extracts the personalized visual features from food images according to individual users’ different visual preferences on them. Given a user-item pair (u, i) , PiNet first uses an image encoder $\mathcal{E}(\cdot)$ to extract the visual features of i that contain both the semantic and collaborative information. Subsequently, the dual-gating module (DGM) re-attends the visual features based on the user embedding p_u to obtain the item embedding q_i . Finally, the matching score \hat{y}_{ui} is computed according to conventional collaborative filtering algorithms, such as MF-BPR.

4 APPROACH

PiNet introduces a heterogeneous multi-task learning framework to learn the visual features of images that fuse both the semantic and collaborative information. As illustrated in Figure 2, PiNet uses an image encoder $\mathcal{E}(\cdot)$ to extract the visual features v_i . Subsequently, the dual-gating module (DGM) learns the task-aware features l_i and q_i for ingredient prediction $l_i \mapsto \hat{d}_i$ and food recommendation $(p_u, q_i) \mapsto \hat{y}_{ui}$, respectively, where \hat{d}_i denotes the predicted ingredients. Notably, PiNet uses the ingredients as ground-truth labels solely in the training phase. In the testing phase, the branch of PiNet for ingredient prediction can be excluded. This design follows a learning paradigm termed learning using privileged information (LUPI) [20, 32, 33], which aims to incorporate the external meta-data of the inputs, i.e. the ingredients in our case, in the training phase to regularize the optimization process.

4.1 Heterogeneous Multi-Task Learning

As shown in Figure 2, PiNet includes two channels, where the task channel takes as input the user and item embeddings p_u and q_i to compute the preference score \hat{y}_{ui} for food recommendation, and

the privileged channel receives the item embedding l_i for ingredient prediction. Both channels share the image encoder $\mathcal{E}(\cdot)$ to extract the unified embedding v_i , and DGM learns to generate q_i and l_i from v_i . As such, $\mathcal{E}(\cdot)$ is optimized during training by the gradients from both tasks. This makes the unified embedding v_i encode both the ingredient and collaborative information. Notably, the image encoder $\mathcal{E}(\cdot)$ can be any of the convolutional neural network (CNN) models. PiNet chooses the class of ResNet models due to its efficiency and successful applications in food recognition and ingredient prediction [1, 19, 20].

4.1.1 Ingredient Prediction in Privileged Channel. The task of ingredient prediction aims to optimize the image encoder $\mathcal{E}(\cdot)$ to encode the semantic information of ingredients in food images. As illustrated in Figure 2, PiNet learns three mappings in the privileged channel to fulfill the ingredient prediction task, including learning the unified visual embedding $\mathcal{E}(i) \mapsto v_i$, reshaping for task-aware features $\mathcal{T}_p(v_i) \mapsto l_i$, and ingredient prediction $\delta(l_i) \mapsto \hat{d}_i$, where $\mathcal{E}(\cdot)$ is the image encoder, $\mathcal{T}_p(\cdot)$ is the task gate for ingredient prediction in DGM, $\delta(\cdot)$ is a fully-connected layer, and \hat{d}_i contains the predicted probabilities for ingredients.

Training of the three mappings for ingredient prediction follows the conventional pipeline of multi-label image classification. Given the set of images \mathcal{I} and the binary ingredient indicator vector \mathbf{d}_i for each image i , the model parameters are optimized by the binary cross-entropy (BCE) loss, defined by

$$\mathcal{L}_p = \sum_{i \in \mathcal{I}} \sum_m [d_{i,m} \log \hat{d}_{i,m} + (1 - d_{i,m}) \log (1 - \hat{d}_{i,m})], \quad (1)$$

where $d_{i,m}$ and $\hat{d}_{i,m}$ denote the m -th elements of \mathbf{d}_i and $\hat{\mathbf{d}}_i$, respectively. As observed, Equation (1) measures the consistency in the predictions for both the presence and absence of ingredients.

4.1.2 Food Recommendation in Task Channel. PiNet follows the collaborative filtering approach for visually-aware food recommendation in the task channel. As shown in Figure 2, this includes the learning of user embedding $\text{emb}(u) \mapsto p_u$, task-aware item embedding $\mathcal{T}_r(v_i) \mapsto q_i$, and scoring function $\hat{y}_{ui} = \mathcal{G}(u, i)$ based on the user and item embeddings p_u and q_i , where $\text{emb}(\cdot)$ is an

action to select the user embedding and $\mathcal{T}_r(\cdot)$ is the task gate for food recommendation in DGM. Training of these mappings follows the widely-used BPR method [25], defined by

$$\mathcal{L}_r = \sum_{(u,i,j) \in \mathcal{D}} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (2)$$

where $\sigma(\cdot)$ is the Sigmoid function and $\mathcal{D} = \{(u, i, j) | i \in \mathcal{I}_u^+, j \in \mathcal{I} \setminus \mathcal{I}_u^+\}$ denotes the set of pairwise training samples. As observed in Equation (2), given a user u and a positive sample i that u has interacted with, BPR selects a negative sample j unseen to u and constrains that the preference score \hat{y}_{ui} should be larger than \hat{y}_{uj} .

There are two commonly-used methods to compute \hat{y}_{ui} . The first method as used in BPR-MF computes the inner product of \mathbf{p}_u and \mathbf{q}_i to measure their “compatibility”, defined by

$$\hat{y}_{ui} = \alpha + \beta_u + \beta_i + \mathbf{p}_u^T \mathbf{q}_i, \quad (3)$$

where α is a global offset, and β_u and β_i are the bias terms. While the second one as used in VBPR additionally model the latent features \mathbf{a}_u and \mathbf{b}_i for u and i , respectively. It is defined by

$$\hat{y}_{ui} = \alpha + \beta_u + \beta_i + \mathbf{p}_u^T \mathbf{q}_i + \mathbf{a}_u^T \mathbf{b}_i. \quad (4)$$

In this case, PiNet as shown in Figure 2 can be extended to incorporate two additional mappings $emb_a(u) \mapsto \mathbf{a}_u$ and $emb_b(i) \mapsto \mathbf{b}_i$.

4.2 Dual-Gating Module

Training a single joint model to simultaneously fulfill the ingredient prediction and food recommendation tasks should handle the heterogeneity in the visual information required by them, as illustrated in Figure 1. To address this issue, PiNet employs the dual-gating module (DGM), which uses the task and gradient gates to enable the image encoder to encode and pass the information required by the respective tasks.

4.2.1 Task Gates. The task gates $\mathcal{T}_p(\cdot)$ and $\mathcal{T}_r(\cdot)$ reshape the unified embeddings \mathbf{v}_i of food image i to generate the task-aware features for ingredient prediction and food recommendation, respectively. In common, both $\mathcal{T}_p(\cdot)$ and $\mathcal{T}_r(\cdot)$ aim to learn a gating vector that can filter out the information in \mathbf{v}_i that is irrelevant to their respective tasks. In contrast, $\mathcal{T}_r(\cdot)$ introduces the user embedding \mathbf{p}_u to make the feature reshaping process personalized.

As shown in Figure 3, the task gate for ingredient prediction $\mathcal{T}_p(\cdot)$ contains the gating and normalization layers to compute the item embedding \mathbf{l}_i for ingredient prediction, defined by

$$\hat{\mathbf{g}}_p = \delta(\text{concat}(\mathbf{v}_i, \mathbf{g}_p)), \quad (5)$$

$$\mathbf{l}_i = \phi(\mathbf{v}_i \odot \hat{\mathbf{g}}_p), \quad (6)$$

where \mathbf{g}_p is a trainable embedding and $\hat{\mathbf{g}}_p$ is the gating vector. $\text{concat}(\cdot)$ performs a concatenation of feature vectors. δ is a fully-connected layer followed by a Sigmoid function. \odot performs element-wise vector product. $\phi(\cdot)$ performs feature normalization to make $\|\mathbf{l}_i\|_2 = \|\mathbf{v}_i\|_2$ where $\|\cdot\|_2$ is the ℓ_2 norm. This operation aims to retain the feature norm after gating and amplify the key features.

The users’ personalized visual preferences are encoded in the task gate for food recommendation $\mathcal{T}_r(\cdot)$, achieved by incorporating the user embedding \mathbf{p}_u to learn the gating vector, defined by

$$\hat{\mathbf{g}}_r = \delta(\text{concat}(\mathbf{p}_u, \mathbf{v}_i, \mathbf{g}_r)), \quad (7)$$

$$\mathbf{q}_i = \theta(\phi(\mathbf{v}_i \odot \hat{\mathbf{g}}_r)), \quad (8)$$

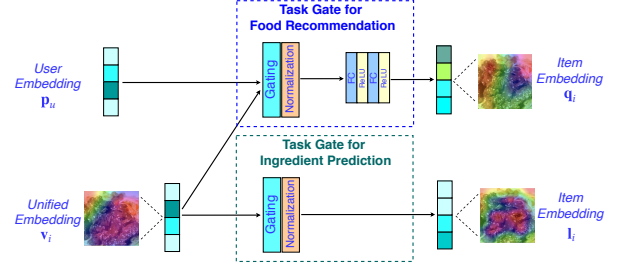


Figure 3: Illustration of task gates that reshape the unified embedding to obtain the task-aware embeddings.

where \mathbf{g}_r and $\hat{\mathbf{g}}_r$ are the trainable and gating vectors, respectively. $\theta(\cdot)$ is a two-layer fully-connected network and each layer is followed by a LeakyReLU activation function. As observed, $\mathcal{T}_r(\cdot)$ allows PiNet to look at different regions of a food image for different users. During training, $\mathcal{T}_r(\cdot)$ is optimized to find the shared visual features among the positive items of individual user u to make their embeddings to be close in the feature space.

4.2.2 Gradient Gate. Fusing the gradients from heterogeneous tasks to optimize the image encoder $\mathcal{E}(\cdot)$ may incur the problem of flat gradients since the tasks may need different visual information. This leads to the trade-off for $\mathcal{E}(\cdot)$ between encoding the ingredient and collaborative information. To address this issue, PiNet employs the gradient gate $\mathcal{F}(\cdot)$, which uses a policy gradient approach to predict the influence of the gradients from \mathcal{L}_r and filter those that strongly go against the gradients from \mathcal{L}_p .

As shown in Figure 4, when learning from the t -th batch of data, the gradient gate $\mathcal{F}(\cdot)$ monitors the action value $\mathbf{s}^{(t-1)} \in [0, 1]$ for the $(t-1)$ -th batch, the values and their changes of the ingredient prediction loss \mathcal{L}_p for the $(t-1)$ -th and t -th batches, i.e. $\mathcal{L}_p^{(t-1)}$, $\mathcal{L}_p^{(t)}$, with $\Delta \mathcal{L}_p^{(t-1)} = \mathcal{L}_p^{(t-1)} - \mathcal{L}_p^{(t-2)}$ and $\Delta \mathcal{L}_p^{(t)} = \mathcal{L}_p^{(t)} - \mathcal{L}_p^{(t-1)}$. Subsequently, an action $\mathbf{s}^{(t)}$ is taken based on them to compute the fused gradients to optimize $\mathcal{E}(\cdot)$, defined by

$$\Theta^{(t+1)} \leftarrow \Theta^{(t)} - \eta(\nabla_{\Theta} \mathcal{L}_p^{(t)} + \mathbf{s}^{(t)} \nabla_{\Theta} \mathcal{L}_r^{(t)}), \quad (9)$$

where η is the learning rate, $\Theta^{(t)}$ is the set of model parameters of $\mathcal{E}(\cdot)$ at the t -th batch, and $\nabla_{\Theta} \mathcal{L}_p$ and $\nabla_{\Theta} \mathcal{L}_r$ are the gradients computed from \mathcal{L}_p and \mathcal{L}_r , respectively.

Specifically, the action $\mathbf{s}^{(t)}$ is computed via a classifier $\pi(\cdot)$, which is a fully-connected layer followed by a Softmax activation function. $\pi(\cdot)$ maps the 5D state vector of $\mathbf{s}^{(t-1)}$ and the monitored loss values to a 5D action space $\mathbf{h} = [h_1, \dots, h_5]$, defined by

$$\mathbf{h} = \pi([\mathcal{L}_p^{(t-1)}, \Delta \mathcal{L}_p^{(t-1)}, \mathbf{s}^{(t-1)}, \mathcal{L}_p^{(t)}, \Delta \mathcal{L}_p^{(t)}]). \quad (10)$$

The action $\mathbf{s}^{(t)}$ is then chosen from $\boldsymbol{\psi} = [0, 0.2, 0.5, 0.8, 1]$ by

$$\mathbf{s}^{(t)} = \boldsymbol{\psi}_{\hat{k}}, \quad \hat{k} = \arg \max_k h_k. \quad (11)$$

By defining a reward function $J_s^{(t)} = \exp(-\mathcal{L}_p^{(t+1)})$ to penalize the increase in the ingredient prediction loss, $\pi(\cdot)$ is optimized by maximizing the probability of $\mathbf{s}^{(t)}$ for the highest $J_s^{(t)}$, defined by

$$\mathcal{L}_{grad} = -\log \sigma(\mathbf{s}_{max}^{(t)} \cdot J_s^{(t)}), \quad (12)$$

where $\mathbf{s}_{max}^{(t)}$ is the probability of $\mathcal{F}(\cdot)$ to select $\mathbf{s}^{(t)}$ and $\sigma(\cdot)$ is the Sigmoid function as used in Equation (2).

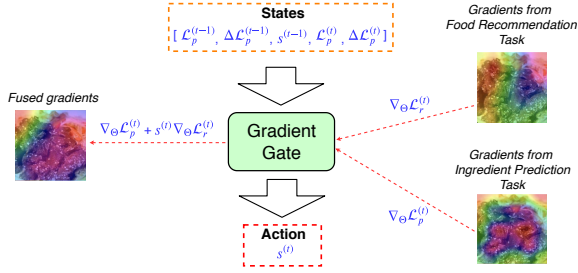


Figure 4: Illustration of gradient gate that uses gradient policy methods to fuse the gradients from heterogeneous tasks.

4.3 Training Strategies for PiNet

Training PiNet from scratch may incur the difficulty in model convergence in the early training stage. This is mainly because that the image encoder $\mathcal{E}(\cdot)$ may be optimized in different directions by gradients from heterogeneous tasks. This may incur the problem that $\mathcal{E}(\cdot)$ cannot encode all the required visual information into the unified embedding \mathbf{v}_i , making the generated item embeddings \mathbf{q}_i and \mathbf{l}_i ill-posed. To address this problem, PiNet incorporates the following methods to ensure an effective model training.

4.3.1 Two-Phase Training. Understanding that the difficulty in model convergence is partially caused by the ineffective learning of $\mathcal{E}(\cdot)$ due to heterogeneous tasks, the two-phase training strategy trains the branches of PiNet in the task and privileged channels independently in the first phase to enable their effective convergence. The second phase then fine-tunes the entire model by jointly training the two branches, as detailed below:

- (1) **Phase I:** Training the two model branches independently is achieved by eliminating the gradients passed from \mathcal{L}_r to optimize the model branch in the privileged channel, i.e. disabling $\mathcal{F}(\cdot)$ with $s^{(t)} = 0$ in Equation (9). In this case, $\mathcal{E}(\cdot)$ and $\mathcal{T}_p(\cdot)$ are optimized by \mathcal{L}_p ; $\mathcal{G}(\cdot)$ and $\mathcal{T}_r(\cdot)$ are optimized by \mathcal{L}_r .
- (2) **Phase II:** The two model branches are then jointly trained by allowing the gradients $\nabla_{\Theta} \mathcal{L}_r$ to pass to the privileged channel to optimize $\mathcal{T}_r(\cdot)$ and $\mathcal{E}(\cdot)$. Specifically, the entire model is iteratively optimized by \mathcal{L}_p and \mathcal{L}_r on different batches of training data to avoid flat gradients. Besides, the gradient gate $\mathcal{F}(\cdot)$ is optimized by \mathcal{L}_{grad} to learn the action value $s^{(t)}$.

4.3.2 Prior Knowledge Regularization. Encoding collaborative information inevitably results in the loss of ingredient information. This may make the image encoder $\mathcal{E}(\cdot)$ lose its focus on the regions of key ingredients to compromise the diverse content in images. To alleviate this problem, PiNet employs a prior knowledge regularization (PKR) method. It adds a loss \mathcal{L}_{prior} to encourage the encoding of ingredient information in \mathbf{l}_i in Phase II, defined by

$$\mathcal{L}_{prior} = \|\mathbf{l}_i - \hat{\mathbf{l}}_i\|_2. \quad (13)$$

where $\hat{\mathbf{l}}_i$ is the item embedding for ingredient prediction in Phase I. Therefore, the final loss for ingredient prediction is $\mathcal{L}_p + \mathcal{L}_{prior}$.

4.3.3 Prior Knowledge Fusion. The prior knowledge fusion (PKF) strategy aims to complement PKR by fusing the ingredient information of $\hat{\mathbf{l}}_i$ into \mathbf{l}_i . More importantly, PKF serves as the second source for \mathbf{l}_i to encode the ingredient information, and it may encourage

Table 1: Statistics of the datasets used in the experiments.

Datasets	#Interactions	#Users	#Items	#Ingredients
Allrecipes	1,093,845	68,768	45,630	2,736
MehishiChina	1,420,723	76,490	61,072	4,628

the unified embedding \mathbf{v}_i to encode more collaborative information to enhance the personalization of item embedding \mathbf{q}_i for food recommendation. The fusion operation is defined by

$$\bar{\mathbf{l}}_i = \text{ReLU}(\mathbf{l}_i + \hat{\mathbf{l}}_i \odot \mathbf{g}_{pkf}), \quad (14)$$

where $\text{ReLU}(\cdot)$ is the ReLU activation function, \mathbf{g}_{pkf} is a trainable gating vector, and \odot performs element product of vectors.

5 EXPERIMENTS

5.1 Experiment Settings

5.1.1 Datasets. Experiments were conducted on two real-world datasets for visually-aware food recommendation. One, called Allrecipes, was built by Gao et al. [9]. It was crawled from Allrecipe.com, a recipe-sharing platform for western food. Notably, this is the only published dataset so far. To better evaluate the generalization capability of PiNet, a new dataset, named MeishiChina, was crawled from meishichina.com for Chinese food recommendation. Statistics of the two datasets are reported in Table 1. In experiments, all of the raw food images were resized to the size of 224x224. The raw ingredients were obtained after data cleaning, including English translation for those from the MeishiChina dataset, converting all uppercase characters to lowercase, removing punctuation and digits, lemmatization, noun extraction, and removing the ingredients that appear only once. Both datasets follow the data partition method as used in [9], where the testing data include the latest 30% of interactions of each user, the training data include the oldest ones of 60%, and the rest of 10% for validation.

5.1.2 Evaluation Protocol. Five commonly-used measures were employed to evaluate the performance of food recommendation, including Precision, Recall, F1 Score, Normalized discounted cumulative gain (NDCG) [9], and AUC [12]. Given a user u and a pair of positive-negative items (i, j) , AUC measures the probability that a recommender obtains $\hat{y}_{ui} > \hat{y}_{uj}$. The other measures compute its performance for the Top-k ranked items, denoted as Precision@k, Recall@k, F1 Score@k, and NDCG@k, respectively. Considering the high dimensionality of image features, negative sampling [9] is used to make the performance evaluation computationally efficient. Specifically, 500 negative items were randomly-sampled from the training set to form the ranking list for each user. To alleviate the issue of randomness, each evaluation was repeated ten times and took the mean value as the final performance.

5.1.3 Implementation Details. PiNet is a model-agnostic framework, so we investigated BPR-MF [25] and VBPR [12] as the base recommenders, denoted as PiNet(BPR-MF) and PiNet(VBPR). Besides, PiNet uses ResNet50 as the base image encoder. This makes all the embeddings used in the privileged channel to have the same size of 2048. In the task channel, the user and item embeddings \mathbf{p}_u and \mathbf{q}_i for BPR-MF and VBPR were set to a range of sizes from {32, 64, 128, 256}. During training, in **training phase I**, the model branch in the privileged channel was optimized by the Adam optimizer with the learning rate set from $1e^{-5}$ to $5e^{-3}$. The weights

Table 2: Performance Comparison between PiNet and existing algorithms on the Allrecipes and MeishiChina datasets. Algorithms are categorized by the methods for food representation. (P@10: Precision@10; R@10: Recall@10; F@10: F1 Score@10)

Food Representation	Algorithms	Allrecipe Dataset					MeishiChina Dataset				
		AUC	P@10	R@10	F@10	NDCG@10	AUC	P@10	R@10	F@10	NDCG@10
Latent Embedding	BPR-MF	0.5329	0.0641	0.2169	0.0849	0.2338	0.5162	0.0588	0.1682	0.0668	0.1769
Pre-Extracted Features	BPR-MF(ResNet50)	0.5629	0.0693	0.2588	0.0887	0.2542	0.5492	0.0630	0.1875	0.0718	0.1816
	VBPR	0.5896	0.0737	0.2653	0.0916	0.2785	0.5712	0.0642	0.1984	0.0737	0.1844
	VECF	0.5980	0.0763	0.2691	0.0931	0.2851	0.5818	0.0648	0.1925	0.0725	0.1877
	HAFR-non-i	0.6062	0.0745	0.2683	0.0942	0.3052	0.5829	0.0654	0.1946	0.0740	0.1885
End-to-End Learning	DVBPR	0.5772	0.0728	0.2668	0.0926	0.2953	0.5613	0.0639	0.1947	0.0729	0.1873
	PiNet(BPR-MF)	0.6097	0.0791	0.2724	0.0967	0.3109	0.5911	0.0672	0.1976	0.0763	0.1928
	PiNet(VBPR)	0.6308	0.0811	0.2776	0.0994	0.3210	0.6057	0.0688	0.2068	0.0787	0.1984
Improvement of PiNet(VBPR) over the Best Baseline Method		4.06%	6.29%	3.15%	5.52%	5.17%	3.91%	5.20%	4.23%	6.35%	5.25%

for \mathcal{L}_p and \mathcal{L}_{prior} were ranged from 1:1 to 1:0.001. The model in the task channel was optimized by the Adagrad optimizer with the learning rate set from $1e^{-4}$ to $5e^{-2}$. The learning rates for both optimizers were multiplied by 0.1 for every four epochs. The batch size was selected from {32, 64, 128, 256}. In **training phase II**, the weights for \mathcal{L}_p and \mathcal{L}_r were ranged from 5:1 to 1:10. Gradient gate $\mathcal{F}(\cdot)$ with \mathcal{L}_{grad} was optimized by the Adam optimizer with the learning rate set from $1e^{-5}$ to $5e^{-3}$.

5.2 Comparison with State-of-the-Art Methods

This section reports the experimental performance of PiNet and six baseline algorithms for food recommendation, including BPR-MF [25], BPR-MF(ResNet50), VBPR [12], VECF [4], HAFR-non-i [9], and DVBPR [14]. BPR-MF(ResNet50) replaces the latent item embedding with the pre-extracted visual features. For a fair comparison, all algorithms used ResNet50 to extract the visual features. The hyperparameters of PiNet and all the baselines were tuned to obtain the best performance by following Section 5.1.3. From the performance as reported in Table 2, we can observe the followings:

- BPR-MF using solely the latent embeddings for food items obtains the worst performance on all performance measures.
- By using the pre-extracted visual features, BPR-MF(ResNet50) usually achieves an increase in performance of 5-10% on all performance measures as compared with BPR-MF. This verifies the importance of visual information in food recommendation.
- BPR-MF(ResNet50) performs the worst among all the algorithms using pre-extracted visual features on all performance measures. The reason lies in that the others additionally model the latent embeddings for users and items. VECF and HAFR-non-i usually outperform VBPR. This is mainly because they employ attention modules to learn the user-aware visual features.
- DVBPR obtains a competitive performance to VBPR on all performance measures. This verifies that both the ingredient and collaborative information signals of food images are important to food recommendation.
- Both PiNet(BPR-MF) and PiNet(VBPR) outperform their base recommenders and the state-of-the-art methods on all performance measures. This verifies that PiNet is able to learn effective visual features for recommendation by fusing the ingredient and collaborative information of food images.
- PiNet(VBPR) consistently outperforms PiNet(BPR-MF). This indicates that the collaborative information discovered from food images may not be sufficient to represent the food items for recommendation. Therefore, modeling the latent embeddings for users and items leads to further improvement in performance.

5.3 Ablation Study

This section explores the influence of various components on the recommendation performance of PiNet. From Table 3(a), the following observations can be drawn:

- **Two-phase training enables effective training of PiNet:** As observed, “Base” yields the worst performance in all cases by training the joint model from scratch. “Base+Pretrain” improves it by the initialization with the pretrained image encoder and recommender. “Base+C” consistently outperforms “Base+Pretrain” and the base recommenders using the pre-extracted ResNet50 features. This verifies that the two-phase training can learn more effective base image encoders and recommenders in training phase I to facilitate the following joint training process.
- **Task gates learn effective task-aware features for food recommendation:** By adding the task gates, “Base+C+TG” consistently improves the recommendation performance of “Base+C” by a large margin in all cases. This is mainly because that learning the task-aware features enables PiNet to better control the trade-off between encoding the ingredient and collaborative information by giving a weight to \mathcal{L}_r higher than that of \mathcal{L}_p .
- **Incorporating the prior knowledge of ingredient information improves the recommendation performance:** Adding either “PR” or “PF” can improve the performance of “Base+C+TG” on all the measures. This verifies the feasibility of introducing prior knowledge to alleviate the loss of ingredient information encoded in I_i . “Base+C+TG+PR” usually outperforms “Base+C+TG+PF” since “PF” does not regularize the learning of I_i . Notably, incorporating both makes “Base+C+TG+PR+PF” further improves the recommendation performance in all cases.
- **Gradient gate alleviates the trade-off between encoding the ingredient and collaborative information:** Incorporating “GG” into “Base+C+TG+PR+PF” leads to consistent improvements for food recommendation in all cases. This indicates that, by limiting the gradients from the recommendation loss \mathcal{L}_r to optimize the image encoder $\mathcal{E}(\cdot)$, “GG” may alleviate the learning from noisy collaborative signals, which go against the semantic ones and may result in a lower performance.

5.4 In-depth Model Analysis

5.4.1 Evaluation of Ingredient Prediction Performance. Following the ablation study of PiNet for food recommendation, this section evaluates the influence of PiNet on ingredient prediction, as illustrated in Table 3(b). The commonly-used Precision@1 [1, 3] is used to evaluate how well PiNet makes the correct prediction for the

Table 3: The performance in (a) food recommendation and (b) ingredient prediction of PiNet(BPR-MF) and PiNet(VBPR) with different combinations of components on the Allrecipes (AR) and MeishiChina (MC) datasets. Base: Plain multi-task model without any components; Pretrain: Initialization with pretrained models; C: Two-phase training; TG: Task gates; PR: Prior knowledge regularization; PF: Prior knowledge fusion; GG: Gradient gate.

Components \ Datasets	PiNet(BPR-MF)		PiNet(VBPR)	
	AR	MC	AR	MC
Base	0.2482	0.1668	0.2551	0.1836
Base+Pretrain	0.2547	0.1834	0.2614	0.1947
Base+C	0.2623	0.1896	0.2669	0.2006
Base+C+TG	0.2664	0.1935	0.2697	0.2024
Base+C+TG+PR	0.2673	0.1948	0.2725	0.2037
Base+C+TG+PF	0.2667	0.1954	0.2708	0.2034
Base+C+TG+PR+PF	0.2685	0.1961	0.2752	0.2046
Base+C+TG+PR+PF+GG (PiNet)	0.2724	0.1976	0.2776	0.2068
BPR-MF/VBPR with ResNet50 Features	0.2588	0.1875	0.2653	0.1984

(a) Performance in Food Recommendation Measured by Recall@10

Components \ Datasets	PiNet(BPR-MF)		PiNet(VBPR)	
	AR	MC	AR	MC
Base	0.0428	0.1872	0.2934	0.3151
Base+Pretrain	0.6471	0.5846	0.6487	0.5863
Base+C	0.6557	0.5969	0.6565	0.6069
Base+C+TG	0.6542	0.5952	0.6558	0.6003
Base+C+TG+PR	0.6563	0.5968	0.6579	0.6125
Base+C+TG+PF	0.6567	0.5971	0.6588	0.6246
Base+C+TG+PR+PF	0.6572	0.5983	0.6604	0.6192
Base+C+TG+PR+PF+GG (PiNet)	0.6583	0.6023	0.6628	0.6211
ResNet50	0.6515	0.5939	0.6515	0.5939

(b) Performance in Ingredient Prediction Measured by Precision@1

Top-1 ranked ingredient. As observed, “Base” and “Base+Pretrain” perform worse than “ResNet50”. This verifies that encoding collaborative information makes the image encoder lose the visual information for ingredient prediction. PiNet alleviates it using the two-phase training strategy, making “Base+C” outperform “ResNet50”. Notably, adding task gates “TG” lowers the performance of “Base+C” since a higher weight for \mathcal{L}_r than \mathcal{L}_p is used to obtain the best recommendation performance. To address this, PiNet employs “PR” and “PF” to encourage the encoding of ingredient information in the task-aware embedding \mathbf{l}_i , and it leads to an improved performance. More importantly, PiNet incorporates the gradient gate “GG” to control the optimization of the image encoder by the collaborative signals. This leads to further improvement in all cases.

5.4.2 Evaluation of Task Gates. This section provides an analysis on the working mechanism of task gates. Figure 5(a) illustrates the feature distributions of the gating vectors $\hat{\mathbf{g}}_p$ and $\hat{\mathbf{g}}_r$ for the task gates $\mathcal{T}_p(\cdot)$ and $\mathcal{T}_r(\cdot)$ of PiNet(BPR-MF). As observed, the gating vector for ingredient prediction $\hat{\mathbf{g}}_p$ does not significantly change the distribution of the unified embedding \mathbf{v}_i due to the normalization procedure in $\mathcal{T}_p(\cdot)$. This indicates that most of the information encoded in \mathbf{v}_i is useful for ingredient prediction. In contrast, $\hat{\mathbf{g}}_r$ behaves diversely to reshape \mathbf{v}_i , indicating a significant difference between the information required for food recommendation and ingredient prediction. The behaviors of the task gates can be explained by the proposed two-phase training method. Since the optimization of $\hat{\mathbf{g}}_r$ happens only in Phase II, the fine-tuning of model will encourages the item embedding \mathbf{l}_i to retain its feature distribution for ingredient prediction; while the task gate $\mathcal{T}_r(\cdot)$ will be optimized to filter useful information in \mathbf{v}_i for food recommendation.

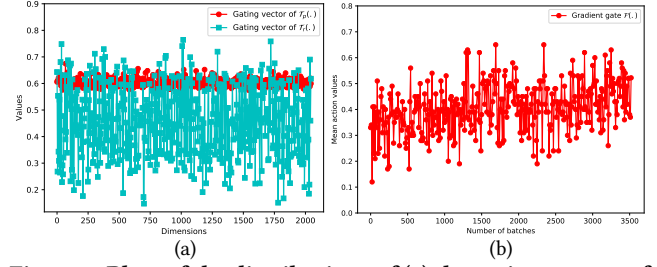


Figure 5: Plots of the distributions of (a) the gating vectors of task gates and (b) the action values taken by gradient gate.

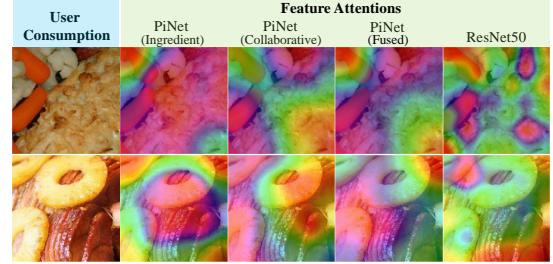


Figure 6: Visualization of feature attentions indicate that PiNet and ResNet50 encode different visual information.

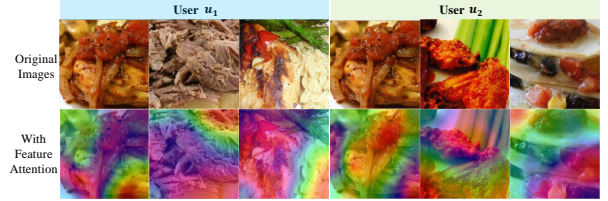


Figure 7: Visualization of personalized feature attentions from PiNet for food recommendation for different users.

5.4.3 Evaluation of Gradient Gate. This section investigates how the gradient gate $\mathcal{F}(\cdot)$ works during training by analyzing the action values s for each batch in the first epoch of training phase II. As shown in Figure 5(b), the strong fluctuation in the action values is caused by the random selection of batches. At the beginning of the training process, the mean value of s approximates 0.3, indicating $s = 0.2$ in most cases. This is because that the encoding of collaborative information leads to significant changes in the parameters of the image encoder $\mathcal{E}(\cdot)$ and the ingredient prediction loss \mathcal{L}_p . Along with the training process, the mean action value keeps increasing, indicating the convergence of the model. Notably, $\mathcal{F}(\cdot)$ still reduces half of the gradients from the recommendation loss \mathcal{L}_r . This indicates that such visual information required for food recommendation does not contribute to ingredient prediction. Therefore, decreasing such gradients helps with the stable model training and leads to performance gains for both tasks.

5.5 Case Studies

5.5.1 Visualization of Ingredient and Collaborative Information encoded by PiNet. This section investigates the ingredient and collaborative information encoded in the unified embedding \mathbf{v}_i extracted by PiNet. This is achieved by using the Grad-CAM [27] to localize the feature attentions of \mathbf{v}_i in the image based on the gradients from the losses of the ingredient prediction and food recommendation tasks, respectively. Figure 6 shows such feature attentions





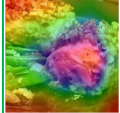
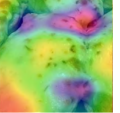
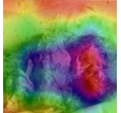
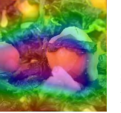

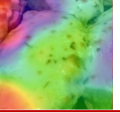
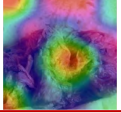

Ground Truth		chopped sugar butter cheese cream	chicken boneless bean vegetable walnut		pepper salt chopped onion butter	cheese chicken tomato breast avocado		pepper salt onion garlic powder	parsley oregano roast lamb rump		salt chopped onion oil garlic	bean cumin spinach garbanzo
Inference of ResNet50		butter chicken cheese boneless parsley	0.936 0.918 0.882 0.844 0.767		honey shrimp potato parsley spaghetti	0.979 0.916 0.808 0.774 0.761		pepper salt lamb garlic parsley	0.915 0.884 0.836 0.822 0.799		grain chopped cubed spinach pastry	0.941 0.935 0.854 0.773 0.757
Inference of PiNet		chicken cheese boneless bean chopped	6.211 5.924 5.811 5.738 5.670		pepper salt cheese chicken butter	5.969 5.764 5.722 5.405 5.110		pepper lamb chopped powder roast	5.529 5.396 5.204 5.191 5.098		garlic pepper bean spinach cheese	5.794 5.468 5.339 5.224 5.161
	(a)			(b)			(c)			(d)		

Figure 8: Error analysis of ingredient prediction using pretrained ResNet50 and PiNet. (a) Both models achieve reasonable performance; (b) ResNet50 fails; (c) PiNet performs worse; and (d) both models perform badly. Bold terms are correct predictions.

generated by PiNet(BPR-MF) for two food images of the same user. As observed, ingredient information typically attends to the major ingredient regions in the images. In contrast, collaborative information attends to the visual characteristics shared by the images, i.e. the red and yellow content. In particular, feature attentions of the bottom image mainly focus on the pineapple regions. We reckon that it is due to the large egg regions in the upper image. Besides, the fused information usually covers the attention regions of both the ingredient and collaborative information. More importantly, comparing with the feature attentions generated by pretrained ResNet50, those of PiNet typically attend to broader regions. This may be a reason for the improved performance of PiNet in both the ingredient prediction and food recommendation tasks.

5.5.2 Personalized Visual Attentions for Food Recommendation. This section evaluates PiNet’s capability to learn the personalized item embedding \mathbf{q}_i for food recommendation. In a similar way as done in Section 5.5.1, the feature attentions generated by PiNet(BPR-MF) for the food images of two users are shown in Figure 7. As observed, for the first image from left shared by u_1 and u_2 , PiNet attends to different regions by considering the users’ different food consumption. Specifically, the feature attentions for the images of u_1 mainly cover the regions of meat and red content. Interestingly, the middle image of beef does not share much with the others in colors, so its attention regions are mainly on the texture of beef and the yellow content at bottom. In contrast, the feature attentions on the shared image of u_2 are mainly on the red tomato regions, since those of the others are mainly on the content with similar color and shapes. These observations verify that PiNet can capture the personalized visual elements of food images for individual users.

5.5.3 Error Analysis of Ingredient Prediction. This section presents an error analysis to investigate how the encoded collaborative information helps with ingredient prediction. As shown in Figure 8, the feature attentions of the item embedding \mathbf{I}_i generated by PiNet(BPR-MF) and the corresponding predicted ingredients with confident scores are compared with those of ResNet50. To summarize, visual features extracted by PiNet usually attend to broader image regions and could better detect both the key and minor ingredients. Specifically, as shown in Figure 8(a), when the ingredients can be clearly distinguished, both models could attend to the regions of key ingredients. Notably, PiNet better attends to and detects

the ingredient of “bean”. Figure 8(b) depicts that when the key ingredients are invisible, ResNet50 fails to attend to the correct regions; while PiNet attends to the pizza-like regions and makes the correct detections. Moreover, as shown in Figure 8(c), although ResNet50 achieves a better performance in Top-5 detection, the encoded collaborative information helps PiNet to correctly detect “roast” and “powder”. Figure 8(d) illustrates a case when some of the ingredients over-occupy the image. Both models attend to the regions of “spinach”, but they fail to predict the key ingredients, such as “garbanzo”. However, PiNet better attends to the central image region and detects “bean” and “garlic”. These observations verify the effectiveness of PiNet for ingredient prediction.

5.6 Conclusion

This paper presents a privileged-channel infused network (PiNet) for visually-aware food recommendation. Conventional methods typically use the pre-extracted visual features. However, such features typically attend to ingredient-intensive regions and cannot capture users’ personalized visual preferences. PiNet addresses this issue by learning the visual features to fulfill both the ingredient prediction and food recommendation tasks. These features therefore encode both the ingredient and collaborative information required by these tasks. Experimental results show that PiNet is able to attend to the image regions of both the ingredients and visual elements shared by the images of individual users. The broader attention regions on informative food content make PiNet outperform existing methods for visually-aware food recommendation.

Future work of this study may focus on two directions. First, PiNet may incorporate a knowledge graph of recipe-ingredient relations and use the predicted ingredients to model users’ food preferences at the semantic level. Second, PiNet can leverage the rich multi-modal recipe information available on the web, such as the cooking procedures, to further enhance current research.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative and the National Natural Science Foundation of China (U19A2079). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

- [1] Marc Bolaños, Aina Ferrà, and Petia Radeva. 2017. Food ingredients recognition through multi-label learning. In *International Conference on Image Analysis and Processing*. Springer, 394–402.
- [2] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang. 2019. Collaborative Similarity Embedding for Recommender Systems. In *The World Wide Web Conference*. ACM, 2637–2643.
- [3] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 32–41.
- [4] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Hongyuan Zha, and Zheng Qin. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multi-model Attention Network. In *SIGIR*, 1–10.
- [5] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Mohan Kankanahalli. 2019. MMALFM: Explainable Recommendation by Leveraging Reviews and Images. *ACM Transactions on Information Systems* 37, 2 (2019), 16:1–16:28.
- [6] Wei-Ta Chu and Ya-Lun Tsai. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web* 20, 6 (2017), 1313–1331.
- [7] David Elswiler, Christoph Trattner, and Morgan Harvey. 2017. Exploiting food choice biases for healthier recipe recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*. ACM, 575–584.
- [8] Jill Freyne and Shlomo Berkovsky. 2010. Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 321–324.
- [9] 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 Transactions on Multimedia* In press (2019), 1–12.
- [10] Mouzhi Ge, Mehdi Elahi, Ignacio Fernández-Tobías, Francesco Ricci, and David Massimo. 2015. Using tags and latent factors in a food recommender system. In *Proceedings of the 5th International Conference on Digital Health*. ACM, 105–112.
- [11] Mouzhi Ge, Francesco Ricci, and David Massimo. 2015. Health-aware food recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 333–334.
- [12] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- [13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), 639–648.
- [14] W. Kang, C. Fang, Z. Wang, and J. McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *2017 IEEE International Conference on Data Mining (ICDM)*. 207–216.
- [15] Deborah A Kerr et al. 2016. The connecting health and technology study: a 6-month randomized controlled trial to improve nutrition behaviours using a mobile food record and text messaging support in young adults. *International Journal of Behavioral Nutrition and Physical Activity* 13, 1 (2016).
- [16] Fang-Fei Kuo, Cheng-Te Li, Man-Kwan Shan, and Suh-Yin Lee. 2012. Intelligent menu planning: Recommending set of recipes by ingredients. In *Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities*. ACM, 1–6.
- [17] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 304–312.
- [18] 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.
- [19] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. 2018. Wide-slice residual networks for food recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 567–576.
- [20] Lei Meng, Long Chen, Xun Yang, Dacheng Tao, Hanwang Zhang, Chunyan Miao, and Tat-Seng Chua. 2019. Learning using privileged information for food recognition. In *ACM international conference on Multimedia*. ACM, 1–9.
- [21] Michele Merler, Hui Wu, Rosario Uceda-Sosa, Quoc-Bao Nguyen, and John R Smith. 2016. Snap, Eat, RepEat: a food recognition engine for dietary logging. In *Proceedings of the 2nd international workshop on multimedia assisted dietary management*. ACM, 31–40.
- [22] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P Murphy. 2015. Im2Calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*. 1233–1241.
- [23] Zhaoyan Ming, Jingjing Chen, Yu Cao, Ciarán Forde, Chong-Wah Ngo, and Tat Seng Chua. 2018. Food Photo Recognition for Dietary Tracking: System and Experiment. In *International Conference on Multimedia Modeling*. Springer, 129–141.
- [24] Charles Packer, Julian McAuley, and Arnau Ramisa. 2018. Visually-Aware Personalized Recommendation using Interpretable Image Representations. In *AI for Fashion workshop, held in conjunction with KDD*. 1–4.
- [25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, 452–461.
- [26] Doyen Sahoo, Wang Hao, Shu Ke, Wu Xiongwei, Hung Le, Palakorn Achananunparp, Ee-Peng Lim, and Steven CH Hoi. 2019. FoodAI: Food Image Recognition via Deep Learning for Smart Food Logging. In *KDD*. 2260–2268.
- [27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.
- [28] Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. 2012. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 298–307.
- [29] Christoph Trattner and David Elswiler. 2017. Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 489–498.
- [30] Mayumi Ueda, Mari Takahata, and Shinsuke Nakajima. 2011. User's food preference extraction for personalized cooking recipe recommendation. In *Workshop of ISWC*. 98–105.
- [31] Youri van Pinxteren, Gijs Geleijnse, and Paul Kamsteeg. 2011. Deriving a recipe similarity measure for recommending healthful meals. In *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 105–114.
- [32] Vladimir Vapnik and Rauf Izmailov. 2015. Learning using privileged information: similarity control and knowledge transfer. *Journal of machine learning research* 16, 2023–2049 (2015), 2.
- [33] Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural networks* 22, 5–6 (2009), 544–557.
- [34] Liping Wang, Qing Li, Na Li, Guozhu Dong, and Yu Yang. 2008. Substructure similarity measurement in chinese recipes. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 979–988.
- [35] Suhan Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. 391–400.
- [36] 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 *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (Melbourne, Australia) (CIKM'15)*. ACM, New York, NY, USA, 183–192.
- [37] 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 Transactions on Information Systems* 36, 1, Article 7 (2017), 31 pages.
- [38] Xun Yang, Meng Wang, Richang Hong, Qi Tian, and Yong Rui. 2017. Enhancing person re-identification in a self-trained subspace. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13, 3 (2017), 1–23.
- [39] Xun Yang, Meng Wang, and Dacheng Tao. 2017. Person re-identification with metric learning using privileged information. *IEEE Transactions on Image Processing* 27, 2 (2017), 791–805.
- [40] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1449–1458.