Towards Better Utilization of Multiple Views for Bundle Recommendation

Kyungho Kim
KAIST
Seoul, Republic of Korea
kkyungho@kaist.ac.kr

Sunwoo Kim
KAIST
Seoul, Republic of Korea
kswoo97@kaist.ac.kr

Geon Lee
KAIST
Seoul, Republic of Korea
geonlee0325@kaist.ac.kr

Kijung Shin
KAIST
Seoul, Republic of Korea
kijungs@kaist.ac.kr

Abstract

Bundle recommender systems aim to recommend suitable collections (i.e., bundles) of items to each user, meeting their diverse needs with all-in-one convenience. Typically, they utilize three distinct types of information: user-bundle purchase interactions (U-B view), user-item purchase interactions (U-I view), and bundle-item affiliations (B-I view). Our focus is on better integrating these three perspectives (i.e., views) to deliver more accurate bundle recommendations. Our examination of different role (main or sub-views) combinations of the views reveals two key observations: (1) the best combination varies across target users (i.e., who receive recommendations), and (2) the U-I view is relatively weak as the main role. Driven by these observations, we propose PET, which synergizes the three views through (1) personalized view weighting, (2) U-I view enhancement, and (3) two-pronged contrastive learning. Our extensive experiments demonstrate that PET significantly outperforms existing methods in all popular benchmark datasets. Our code and datasets are available at https://github.com/K-Kyungho/PET.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Bundle Recommendation, Multi-view Fusion, Personalization

ACM Reference Format:
Kyungho Kim, Sunwoo Kim, Geon Lee, and Kijung Shin. 2024. Towards Better Utilization of Multiple Views for Bundle Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3680003

1 Introduction

A *bundle* refers to a set of items that are grouped by a particular concept. Bundles offer convenience to users by providing desirable items all at once. Real-world examples of bundles include playlists in music streaming platforms, fashion outfits in online shopping malls, and combo meals in food delivery services.

Bundle recommender models [1, 3, 4, 6, 10, 15–17, 20, 23], designed to recommend suitable bundles to users, typically leverage three different types of information (or views): (1) user-bundle interactions (U-B view), i.e., who purchases which bundle; (2) user-item interactions (U-I view), i.e., who buys which item; and (3) bundle-item affiliation (B-I view), i.e., the composition of each bundle.

Toward better synergy of the three views, we analyze the real-world bundle recommendation datasets. Specifically, we focus on the combination of views, identifying which view is the most effective as the main role (spec., input for message passing of GNN-based recommendation models). We reveal two key observations:

(O1) The best combination of the views varies across target users.
(O2) The U-I view is the least effective view as the main role.

Based on these observations, we propose a new bundle recommender model, **PET** (**P**ersonalized view weighting with data **E**nhancement **T**wo-pronged contrast), which synergizes the three views through three techniques. First, based on **(O1)**, we utilize view weights personalized to each user when merging views. Second, based on **(O2)**, we enhance the user-item interaction view by using the other two views. Third, we employ two-pronged contrastive learning (CL), combining two CL strategies to improve representation learning. Specifically, in addition to *Inter-CL* (i.e., contrastive learning across different views), which has been demonstrated to be effective in bundle recommendation [16, 23], we employ *Intra-CL*. That is, we create two augmented views from each view and contrast them, enabling PET to learn robust representations even when each view contains only limited interactions. We also validate the effectiveness of PET through extensive experiments.

Our contributions are summarized as follows:

- **Observation.** We reveal two intriguing observations regarding the integration of multiple views in real-world bundle datasets.
- **Method.** We propose PET, a novel bundle recommendation model that achieves enhanced integration of multiple views.
- **Experiments.** PET achieves up to a 39.26% performance gain compared to the best competitors.

2 Related Work & Preliminaries

In this section, after briefly reviewing related studies, we provide preliminaries and formulation of the bundle recommendation task.

2.1 Related Work

GNN-based recommendation models A significant focus has been placed on graph-neural-network (GNN) encoders and their training strategies for effective recommendation. Among the encoders, LightGCN [9] stands out due to its simplicity and superior performance across various domains [12, 16, 19, 21, 22].

GNN-based bundle recommendation models. GNN-based bundle recommendation models typically leverage three types of interactions (or views): user-bundle, user-item, and bundle-item. For instance, Deng et al. [6] unified these views into a single graph and employed GCN [14]. Instead of unifying all views, several

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3680003

Table 1: Dataset statistics of real-world bundle datasets.

Dataset	#User	#Item	#Bundle	#U-I	#U-B	#Avg.I/B
iFashion	53,897	42,563	27,694	2,290,645	1,679,708	3.86
NetEase	18,528	123,628	22,864	1,128,065	302,303	77.80
Youshu	8,039	32,770	4,771	138,515	51,377	37.03

studies learned distinct representations (i.e., user/item/bundle embeddings) from each view [3, 16, 17, 23]. For instance, Ma et al. [16] and Zhao et al. [23] contrasted representations obtained from different views to improve representation learning. Recent studies further enhanced the synergy of the views, for instance, by employing early fusion strategies [15], transformer-based attention mechanisms [20], and knowledge distillation [17].

2.2 Notations and Problem Formulation

For a user set $\mathcal{U} = \{u_1, \cdots, u_{|\mathcal{U}|}\}$, an item set $\mathcal{I} = \{i_1, \cdots, i_{|\mathcal{I}|}\}$, and a bundle set $\mathcal{B} = \{b_1, \cdots, b_{|\mathcal{B}|}\}$, matrices $\mathbf{A}^{(UB)} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{B}|}$, $\mathbf{A}^{(UI)} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}$, and $\mathbf{A}^{(BI)} \in \{0,1\}^{|\mathcal{B}|\times|\mathcal{I}|}$ denote the user-bundle (U-B), user-item (U-I), and bundle-item (B-I) interactions, respectively. Each entry $\mathbf{A}^{(UB)}_{ij}$ (or $\mathbf{A}^{(UI)}_{ik}$) is 1 if there exists an interaction between u_i and b_j (resp., u_i and i_k), and 0 otherwise. Similarly, each entry $\mathbf{A}^{(BI)}_{jk}$ is 1 if i_k belongs to b_j, and 0 otherwise. We refer to $\mathbf{A}^{(UB)}$, $\mathbf{A}^{(UI)}$, and $\mathbf{A}^{(BI)}$ as the U-B, U-I, and B-I views, respectively. For any matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, its row-wise normalized version is denoted by $\hat{\mathbf{A}}$ (i.e., $\hat{\mathbf{A}}_{ij} = \mathbf{A}_{ij}/\sum_{k=1}^{m}\mathbf{A}_{ik}$). The objective of bundle recommendation is to find unobserved user-bundle interactions from the given matrices $\mathbf{A}^{(UB)}$, $\mathbf{A}^{(UI)}$, and $\mathbf{A}^{(BI)}$.

3 Real-world Data Analysis

We present our analysis of real-world bundle datasets (spec., iFashion [5], NetEase [2], and Youshu [4]), focusing on the integration of their three views. Some statistics for the datasets are in Table 1.

3.1 Analysis Settings

Key concepts. We outline two key concepts: the main view and the sub view. A main view is the one for which we learn representations (i.e., embeddings) by applying the GNN message passing on it. A sub-view is used to pool the embeddings from the main view to obtain user or bundle embeddings, if necessary. For example, when we adopt the U-B view as the main view, we directly learn user and bundle embeddings, making the sub-view unnecessary. However, when we adopt the U-I (or B-I) view as the main view, we do not learn bundle (resp., user) embeddings from the main view. Instead, we utilize the B-I (resp., U-I) view as the sub-view to obtain bundle (resp., user) embeddings by pooling item embeddings. These user and bundle embeddings are used for bundle recommendation, as detailed below. We aim to analyze the effectiveness of the U-B, U-I, and B-I views as the main view for bundle recommendation.

Details. We split a set of user-bundle interactions into 70%/10%/20% for training/validation/test.[1] For each view, we use LightGCN [9] and compute the score of a user u_i purchasing a bundle b_j via the inner product of their respective embeddings. For training, we employthe BPR loss [18] with training user-bundle interactions as positive pairs and false user-bundle interactions as negative ones.

[1]Note that U-B view consists only of training user-bundle interactions.

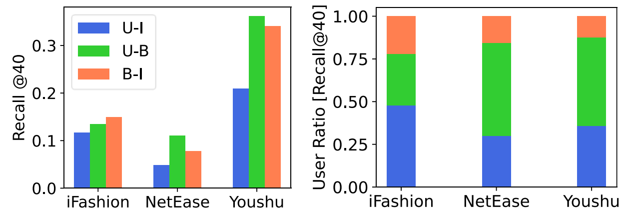

Figure 1: Performance (Recall@40) when utilizing each view as the main view (left) and the proportion of users for whom each main view leads to the best performance (right).

Lastly, we evaluate the scores from each view using test user-bundle interactions in terms of Recall@40 and NDCG@40.[2]

3.2 Analysis Results

We present user- and global-level observations from our analysis.

(O1) One size does not fit all. For each user, we (1) compute Recall@40 and NDCG@40 for each view as the main view, and (2) identify which main view yields the best prediction performance for that user. We then calculate the ratio of users for which each view achieves the best performance. As shown in Figure 1(b), we observe that the main view leading to the best performance varies among different users, with no single view consistently leading to the best performance across the majority of users.

(O2) U-I view is relatively weak. As shown in Figure 1(a), using the U-I view as the main view leads to the lowest performance compared to the other two views.

4 Proposed Method

We propose a model for bundle recommendation, **PET** (**P**ersonalized view weighting with data **E**nhancement **T**wo-pronged contrast). Refer to Figure 2 for a visual description. It employs three techniques that collectively enhance the utilization of multiple views.

(T1) U-I-enhanced view representation. To address the relative weakness of the U-I view (**O2**), we enhance it by integrating extra user-item interactions derived from the other views, as follows:

$$\mathbf{A}^{(UI)'} = \mathbf{A}^{(UI)} + \beta(\mathbf{A}^{(UB)})^T\mathbf{A}^{(BI)} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|},$$

where $\beta \in \mathbb{R}_{\geq 0}$ controls the degree of the view enhancement. The additional U-I view, $(\mathbf{A}^{(UB)})^T\mathbf{A}^{(BI)}$, provides information about the items purchased by each user indirectly through bundles, supplementing the direct user-item interactions. By leveraging the three views, including the enriched U-I view, we obtain user and bundle embeddings from each view as follows:

$$\mathbf{U}^{(UI)},\ \mathbf{I}^{(UI)} = \mathsf{GNN}(\mathbf{A}^{(UI)'}, \mathbf{U}', \mathbf{I}'), \qquad \mathbf{B}^{(UI)} = \hat{\mathbf{A}}^{(BI)}\mathbf{I}^{(UI)},$$
$$\mathbf{B}^{(BI)},\ \mathbf{I}^{(BI)} = \mathsf{GNN}(\mathbf{A}^{(BI)}, \mathbf{B}', \mathbf{I}'), \qquad \mathbf{U}^{(BI)} = \hat{\mathbf{A}}^{(UI)'}\mathbf{I}^{(BI)},$$
$$\mathbf{U}^{(UB)},\ \mathbf{B}^{(UB)} = \mathsf{GNN}(\mathbf{A}^{(UB)}, \mathbf{U}', \mathbf{B}'),$$

where, $\mathbf{U}'$, $\mathbf{I}'$, and $\mathbf{B}'$ are the learnable initial embeddings of users, items, and bundles, respectively, which are shared across all views.

(T2) Personalized view weighting. Noting that no single view is optimal for all users (**O1**), we introduce a personalized view weighting scheme. Specifically, we compute the importance of each view, which is personalized for each user, by considering the dependencies between views as follows:

[2]The results in terms of NDCG@40 are available at [11].

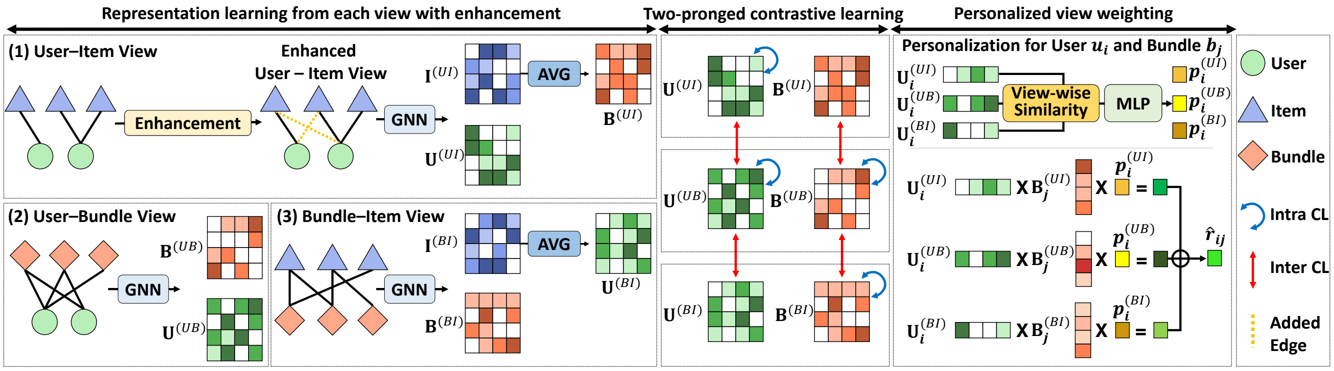

Figure 2: Visualization of PET, the proposed bundle recommendation model. The three key components of PET are (1) user-item view enhancement, (2) two-pronged contrastive learning, and (3) personalized view weighting.

$$p_i = \mathtt{Softmax}\left(\mathtt{MLP}\left(\mathbf{U}_i\mathbf{U}_i^T\right)\right) \in \mathbb{R}^3, \tag{1}$$

where $\mathbf{U}_i = [\mathbf{U}_i^{(UI)}; \mathbf{U}_i^{(UB)}; \mathbf{U}_i^{(\mathrm{BI})}] \in \mathbb{R}^{3\times d}$ represents the stacked embeddings of user u_i from the three views. The product of the embeddings from each pair of views is processed through an MLP, followed by a softmax function. This results in a probability vector $p_i = [p_i^{(UI)}, p_i^{(UB)}, p_i^{(BI)}]$ whose components represent the personalized importances of the views U-I, U-B, and B-I for user u_i. Using these personalized importances, the estimated relevance score $\hat{r}_{ij}$ of a bundle b_j and a user u_i is computed as:

$$\hat{r}_{ij} = \sum\nolimits_{V\in\mathcal{V}} p_i^{(V)} \cdot (\mathbf{U}_i^{(V)})^T \mathbf{B}_j^{(V)}, \tag{2}$$

where $\mathcal{V} = \{UI, UB, BI\}$ is the set of views. The estimated relevance scores from each view are aggregated with their respective importance obtained from the personalized weighting.

(T3) Two-pronged contrastive learning. In practice, we typically have a limited number of user-bundle interactions. To enhance representation learning under these circumstances, we introduce *Intra-CL*, i.e., contrastive learning within each view. Specifically, for user and bundle embeddings, we leverage the following InfoNCE [8]-based contrastive loss for intra-CL:

$$\mathcal{L}_c^{\text{intra}} = -\sum_{V\in\{UI,UB\}}\sum_{i\in\mathcal{U}} \log \frac{\exp\left(\cos\left(\mathbf{U}_i^{(V,1)}, \mathbf{U}_i^{(V,2)}\right)/\tau\right)}{\sum_{i'\in\mathcal{U}}\exp\left(\cos\left(\mathbf{U}_i^{(V,1)}, \mathbf{U}_{i'}^{(V,2)}\right)/\tau\right)} - \sum_{V\in\{BI,UB\}}\sum_{j\in\mathcal{B}} \log \frac{\exp\left(\cos\left(\mathbf{B}_j^{(V,1)}, \mathbf{B}_j^{(V,2)}\right)/\tau\right)}{\sum_{j'\in\mathcal{B}}\exp\left(\cos\left(\mathbf{B}_j^{(V,1)}, \mathbf{B}_{j'}^{(V,2)}\right)/\tau\right)},$$

where $\mathtt{cos}(\mathbf{x}, \mathbf{y})$ is a cosine similarity (i.e., $\mathbf{x}^T\mathbf{y}/(\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2)$), $\tau \in \mathbb{R}_{\geq 0}$ is a hyperparameter, and $\mathbf{U}_i^{(V,1)}$ and $\mathbf{U}_i^{(V,2)}$ (or $\mathbf{B}_j^{(V,1)}$ and $\mathbf{B}_j^{(V,2)}$) are user (resp., bundle) embeddings learned from two different dropout-based graph augmentations of $\mathbf{A}^{(V)}$.[3] Intra-CL encourages consistency between the same user (or bundle) embeddings in two augmented views, while minimizing the agreement of arbitrary pairs of users (resp., bundles).

In addition, we incorporate *Inter-CL* (i.e., inter-view contrastive learning) to align user and bundle embeddings obtained across views. The inter-CL contrastive loss is defined as:

[3] Specifically, we randomly mask q% of ones in $\mathbf{A}^{(V)}$ with zeros.

$$\mathcal{L}_c^{\text{inter}} = -\sum_{V\in\{UI,BI\}}\sum_{i\in\mathcal{U}} \log \frac{\exp\left(\cos\left(\mathbf{U}_i^{(UB,1)}, \mathbf{U}_i^{(V,1)}\right)/\tau'\right)}{\sum_{i'\in\mathcal{U}}\exp\left(\cos\left(\mathbf{U}_i^{(UB,1)}, \mathbf{U}_{i'}^{(V,1)}\right)/\tau'\right)} - \sum_{V\in\{UI,BI\}}\sum_{j\in\mathcal{B}} \log \frac{\exp\left(\cos\left(\mathbf{B}_j^{(UB,1)}, \mathbf{B}_j^{(V,1)}\right)/\tau'\right)}{\sum_{j'\in\mathcal{B}}\exp\left(\cos\left(\mathbf{B}_j^{(UB,1)}, \mathbf{B}_{j'}^{(V,1)}\right)/\tau'\right)},$$

where $\tau' \in \mathbb{R}_{\geq 0}$ is a hyperparameter. Inter-CL encourages the alignment between user (or bundle) embeddings from the U-B view and user (resp., bundle) embeddings from the U-I and B-I views.

The two-pronged contrastive loss is obtained by combining the contrastive losses for the intra-CL and inter-CL as follows:

$$\mathcal{L}_{\text{CL}} = \lambda_1 \cdot \mathcal{L}_c^{\text{intra}} + \lambda_2 \cdot \mathcal{L}_c^{\text{inter}}, \tag{3}$$

where λ_1 and λ_2 are hyperparameters.

Learning objective. In addition to the contrastive loss (i.e., Eq. (3)), we employ the BPR [18] loss based on the personalized user-bundle score (Eq. (2)), using positive and negative user-bundle pairwise interactions. Additionally, we use view-specific BPR losses from each view as an auxiliary loss $\mathcal{L}_{\text{aux}}$ (refer to [11] for more details). Lastly, to prevent a single view from undesirably dominating the importance of all users, we introduce a regularization term, $\mathcal{L}_{\text{reg}} = \sum_{u_i\in\mathcal{U}} \|p_i\|_2$, resulting in the final objective of PET as follows:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \mathcal{L}_{\text{CL}} + \lambda_3\mathcal{L}_{\text{aux}} + \lambda_4\mathcal{L}_{\text{reg}},$$

where λ_3 and λ_4 are hyperparameters. Jointly optimizing these loss terms enhances the performance of PET, as shown in Section 5.

5 Experimental Results

In this section, we review our experiments for three questions:

- **RQ1. Performance comparison.** How effective is PET compared to the state-of-the-art bundle recommender systems?
- **RQ2. Ablation study.** Are all the key components of PET necessary for its performance?
- **RQ3. Effect of personalization.** Are the learned personalization weights of PET diversified, as we intend?

5.1 Experimental Settings

We provide full details regarding the experimental settings in [11].

Datasets and splits. We use the three datasets employed for our data analysis (Section 3). We follow the common train/validation/test splits of user-bundle interactions [15, 16, 20]. In each dataset, we

Table 2: Bundle recommendation performance. The best performance is highlighted in bold, and the second-best one is underlined. In all the cases, PET outperforms all the baseline methods.

Datasets	iFashion				NetEase				Youshu			
Metrics	Recall@20	NDCG@20	Recall@40	NDCG@40	Recall@20	NDCG@20	Recall@40	NDCG@40	Recall@20	NDCG@20	Recall@40	NDCG@40
BPRMF [18]	0.0882	0.0647	0.1347	0.0612	0.0677	0.0363	0.1082	0.0469	0.2660	0.1532	0.3691	0.1835
LightGCN [9]	0.0957	0.0707	0.1439	0.0876	0.0751	0.0397	0.1184	0.0508	0.2750	0.1622	0.3757	0.1859
MIDGN [23]	0.0694	0.0500	0.1091	0.0640	0.0680	0.0358	0.1075	0.0460	0.2688	0.1562	0.3696	0.1836
BundleGT [20]	0.0981	0.0726	0.1471	0.0898	0.0913	0.0481	0.1394	0.0607	0.2927	0.1745	0.3964	0.2028
CrossCBR [16]	0.1308	0.1004	0.1888	0.1209	0.0901	0.0485	0.1372	0.0609	0.2831	0.1689	0.3843	0.1968
MultiCBR [15]	0.1863	0.1569	0.2475	0.1779	0.0928	0.0509	0.1391	0.0631	0.2932	0.1732	0.3968	0.2012
PET	**0.2532**	**0.2185**	**0.3220**	**0.2429**	**0.0978**	**0.0528**	**0.1459**	**0.0655**	**0.3052**	**0.1804**	**0.4103**	**0.2095**
Improvement	35.91%	39.26%	30.10%	36.54%	5.39%	3.73%	4.66%	3.80%	4.09%	3.38%	3.40%	3.30%

Table 3: Effectiveness of the key components of PET. The best performance is highlighted in bold, and the second-best one is underlined. R@40: Recall@40. N@40: NDCG@40.

Datasets	iFashion		NetEase		Youshu	
Metrics	R@40	N@40	R@40	N@40	R@40	N@40
PET-E.	0.3161	0.2420	0.1413	0.0638	0.4081	**0.2092**
PET-P.	0.3215	0.2427	0.1406	0.0636	0.4011	0.2030
PET-I.	0.2917	0.2209	0.1393	0.0629	0.4028	0.2056
PET	**0.3220**	**0.2429**	**0.1459**	**0.0655**	**0.4103**	0.2090

conduct five trials with different model initializations and report the average performance.

Baseline methods. We use six baseline methods with official code released by their authors, including two general recommendation models (BPRMF [18] and LightGCN [9]) and four bundle recommendation models (MIDGN [23], CrossCBR [16], MultiCBR [15], and BundleGT [20]). Note that MultiCBR and BundleGT are the state-of-the-art bundle recommendation models.

Evaluation metrics. We employ NDCG@N and Recall@N, which have been used widely in prior studies [15, 16, 20], to measure the performance of each method. We set $N = 20$ and 40.

Hyperparameters. For all the considered methods, we (1) use the Xavier initialization [7] and the Adam optimizer [13]; (2) fix the learning rate and batch size to 0.001 and 2,048, respectively; and (3) tune the embedding size in $\{64, 128, 256\}$.[4] For PET, we leverage 1-layer LightGCN [9] as the GNN encoder and consider the following hyperparameter search spaces: (1) the augmentation strength in $\{10, 30, 50, 70\}$; (2) the temperature parameters τ and τ' in $\{0.1, 0.2, \cdots, 1.0\}$; (3) the loss coefficients $\lambda_1, \lambda_2, \lambda_3$, and the U-I the enhancement coefficient β in $\{0.01, 0.03, 0.05, 0.07, 0.1, 0.2, 0.3, 0.5, 1, 2\}$; and (4) the regularization coefficient λ_4 in $\{10^{-7}, 10^{-6}, 3\times 10^{-6}, 5\times 10^{-6}, 10^{-5}, 3\times 10^{-5}, 5\times 10^{-5}, 10^{-4}\}$.

5.2 Experimental Results

(RQ1) Performance comparison. We compare the performance of PET against that of the baseline methods. As shown in Table 2, PET outperforms all the baseline methods across all the metrics and datasets, achieving up to 39.26% performance gain compared to the best competitor, in terms of NDCG@20. The performance gain is especially remarkable in the iFashion dataset, where the B-I and U-I views are effective as the main view (see Figure 1). We hypothesize

[4]In the NetEase dataset, BundleGT and MIDGN suffer from an out-of-memory issue when their embedding sizes are 128 or 256. In such cases, we fixed their sizes as 64.

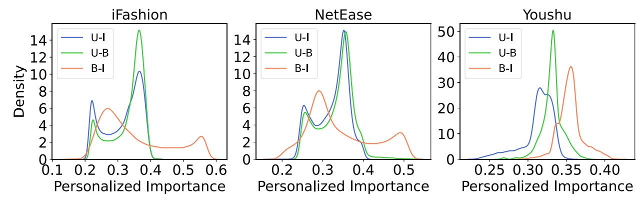

Figure 3: A KDE plot of the distributions of personalized view weights for users. Note that the weights are spread across a wide range, consistent with our design goal.

that Intra-CL enables PET to learn beneficial information from the B-I and U-I views by improving the embeddings from them.

(RQ2) Ablation study. We demonstrate the effectiveness of the key components of PET. To this end, we consider three variants: PET without U-I view enhancement **(PET-E.)**, PET without personalized view weighting **(PET-P.)**, and PET without Intra-CL **(PET-I.)**. As shown in Table 3, PET outperforms its variants in five out of six cases, demonstrating the necessity of each key component of it.

(RQ3) Effect of personalization. As discussed in Section 3, effective main views vary among users, and the personalization weights of PET (Eq (1)) are designed to adaptively prioritize views for each user. As shown in Figure 3, personalization weights for each view are spread across a wide range, specifically from 0.15 to 0.6.[5] This result indicates that the personalization weights indeed vary among users, and thus our design goal is achieved.

6 Conclusions

In this work, we focus on the synergy of the three different types of information (i.e., views) for bundle recommendation, i.e., user-bundle interactions, user-item interactions, and bundle-item affiliation. First, we analyze real-world bundle datasets and reveal key properties of the three views (Section 3). Second, motivated by them, we propose PET, a new bundle recommendation model (Section 4). Lastly, we empirically validate the effectiveness of PET (Section 5). For future work, we plan to extend our research to various recommender systems that can benefit from the synergy of multiple views (e.g., multi-behavior, social, group recommendations).

Acknowledgements. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00438638, EntireDB2AI: Foundations and Software for Comprehensive Deep Representation Learning and Prediction on Entire Relational Databases, 90%) (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST), 10%).

[5]The theoretical bound of the personalization weight is (0, 1).

References

[1] Tzoof Avny Brosh, Amit Livne, Oren Sar Shalom, Bracha Shapira, and Mark Last. 2022. BRUCE : Bundle Recommendation Using Contextualized item Embeddings. In *RecSys*.

[2] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. Embedding Factorization Models for Jointly Recommending Items and User Generated Lists. In *SIGIR*.

[3] Jianxin Chang, Chen Gao, Xiangnan He, and Depeng Jin. 2020. Bundle Recommendation with Graph Convolutional Networks. In *SIGIR*.

[4] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching User with Item Set : Collaborative Bundle Recommendation with Deep Attention Network. In *IJCAI*.

[5] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG:Personalized Outfit Generation for Fashion Recommendation at Alibaba Fashion. In *KDD*.

[6] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. 2020. Personalized Bundle Recommendation in Online Games. In *CIKM*.

[7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.

[8] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.

[10] Zhankui He, Handong Zhao, Tong Yu, Sungchul Kim, Fan Du, and Julian Mcauley. 2022. Bundle MCR : Towards Conversational Bundle Recommendation. In *RecSys*.

[11] Kyungho Kim, Sunwoo Kim, Geon Lee, and Kijung Shin. 2024. Code, Datasets, and Appendix for "Towards Better Utilization of Multiple Views for Bundle Recommendation". https://github.com/K-Kyungho/PET.

[12] Taeri Kim, Yeon-Chang Lee, Kijung Shin, and Sang-Wook Kim. 2022. MARIO: Modality-Aware Attention and Modality-Preserving Decoders for Multimedia Recommendation. In *CIKM*.

[13] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[14] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[15] Yunshan Ma, Yingzhi He, Xiang Wang, Yinwei Wei, Xiaoyu Du, Yuyangzi Fu, and Tat-Seng Chua. 2024. MultiCBR:Multi-view Contrastive Learning for Bundle Recommendation. *ACM Transactions on Information Systems* 42, 100 (2024), 1–23.

[16] Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. 2022. CrossCBR:Cross-view Contrastive Learning for Bundle Recommendation. In *KDD*.

[17] Yuyang Ren, Zhang Haonan, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2023. Distillation-enhanced Graph Masked Autoencoders for Bundle Recommendation. In *SIGIR*.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618 [cs.IR]

[19] Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. 2023. Lightgt: A light graph transformer for multimedia recommendation. In *SIGIR*.

[20] Yinwei Wei, Xiaohao Liu, Yunshan Ma, Xiang Wang, Liqiang Nie, and Tat-Seng Cua. 2023. Strategy-aware Bundle Recommender System. In *SIGIR*.

[21] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *MM*.

[22] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *SIGIR*.

[23] Sen Zhao, Wei Wei, Ding Zou, and Xiangling Mao. 2022. Multi-view Intent Disentangle Graph Networks for Bundle Recommendation. In *AAAI*.