Personalized Outfit Compatibility Prediction Using Outfit Graph Network

B S Vivek, Gaurab Bhattacharya, Jayavardhana Gubbi, Bagya Lakshmi V, Arpan Pal and P. Balamuralidhar
TCS Research, India

***Abstract*—Recommendation systems improve users' online shopping experience by recommending relevant items from a large pool of items in different categories. Fashion recommendation systems apart from recommending individual fashion items also recommend fashion outfits. In this work, we consider the problem of the outfit compatibility prediction task, an integral part of the fashion outfit recommendation system. A compatibility prediction module determines whether all the items in an outfit are visually compatible with each other and match the user's preferences. Existing approaches can be grouped based on the representation scheme: (i) pair-wise and (ii) set or sequence. Pair-wise representation does not consider the outfit as a whole, and the sequence representation approaches are sensitive to the ordering of the items. Further, these methods do not explicitly capture the visual relationship between the items. We propose a novel method for the personalized outfit-compatible prediction task. The proposed method represents the outfit as a graph and uses a dot-attention graph neural network to capture the visual relationship between items. The graph read-out layer generates the final outfit embedding. A novel approach is proposed to model the user's preference for different styles. The final outfit compatibility score is generated by computing the similarity between outfit embedding and user embedding. Experimental results and ablation study on the Polyvore-U dataset, highlight the effectiveness of the proposed method.**

I. Introduction

Recommendation systems (RecSys) play a vital role in e-commerce platforms. A significant percentage of sales on these platforms is based on recommendations [1]. RecSys improve users' shopping experience by helping them find relevant products. Users' feedback and purchase history help RecSys to recommend suitable products to the users. RecSys have been adapted for various business domains, *e.g.*, video streaming services, fashion, grocery, *etc.* In fashion e-commerce platforms, recommendation systems also recommend fashion outfits apart from complementary and substitute products. A fashion outfit is a set of fashion items or apparel that are worn together and represent a particular style.

In this work, we consider the outfit compatibility prediction task. The outfit compatibility prediction module is a critical component of an outfit recommendation system. An outfit is said to be compatible if all the items are visually compatible and match the user's preferences. Various factors such as demography, season, occasion, and user preferences, affect the outfit compatibility score. The outfit compatibility is subjective, *i.e.*, an outfit liked by one user need not necessarily be preferred by another user. The use of population preferences for outfit recommendations leads to sub-optimal results. Therefore, it is necessary to consider the user's preferences for compatibility scoring.

Unlike visual similarity, visual compatibility is a complex concept. Determining visual similarity between fashion items involves comparing visual attributes of the items, *e.g.*, the color of the shirt and trousers. Visual similarity helps in fashion item search and substitute item recommendation. Visual compatibility uses latent concepts to determine the compatibility between fashion items. In the case of fashion outfits, all the items should be visually compatible with each other. Existing methods for outfit compatibility prediction can be classified into two groups, (i) works that predict compatibility between each pair of fashion items [2]–[4], (ii) works that predict compatibility at the outfit level [5], [6]. Pair-wise approaches generate compatibility scores for all pairs of fashion items. The outfit score is generated by aggregating these computed scores. Pair-wise approaches do not consider the outfit as a whole and fail to capture the interactions between items in the outfit. Works such as [5], [6] generate compatibility scores at the outfit level. [5] represents the outfit as a sequence of items and uses LSTM for generating the score (sensitive to the order of items). [6] represents the outfit as a set and uses a set transformer for score generation (does not explicitly model the interaction between items). Some initial works on outfit compatibility prediction discarded user preferences. These methods implicitly model the population preferences. The following are the challenges associated with the outfit compatibility prediction task:

- Variable size: Outfit can have a variable number of fashion items.
- Complex interactions between fashion items. There exist complex interactions between fashion items. Certain items in the outfit are more important compared to others.
- Typically, users have different preferences for different fashion styles (*e.g.,* casual, formal)

To address these challenges, we propose a novel approach for personalized outfit compatibility prediction. The proposed method represents the outfit as a graph and uses dot attention graph neural network (GNN) to capture the inter-relationship between the items. A graph read-out layer generates the final outfit embedding. The proposed approach efficiently models the preferences of the users for different styles. Finally, the outfit compatibility score is generated by computing the similarity between the outfit embedding and the user embedding. Experimental results on the Polyvore-U [2] dataset highlight

the effectiveness of the proposed method. Following are the contributions of our work:

- We propose a novel model for the personalized outfit compatibility prediction task. The proposed method explicitly captures the relationship between the fashion items.
- The proposed user preference modeling stream generates user embeddings for different styles.
- The proposed approach can handle new users without retraining the model.

The paper is organized as follows, Section II discusses related works, Section III describes the proposed approach, Section IV discusses the experimental results, and finally the Section V concludes the paper.

II. Related works

Existing approaches for outfit compatibility prediction can be grouped into two types: (i) methods without personalization and (ii) methods with personalization. We can further classify these methods based on the comparison or matching technique (a) pair-wise comparison, and (b) outfit comparison. Our work falls into the sub-groups of (ii) and (b) outfit comparison with personalization.

A. Outfit compatibility works without personalization

The main aim of the methods in this group is to learn an optimal representation of the items or outfits. McAuley *et al.* [7] proposed Mahalanobis distance transformation to assess product compatibility based on style and appearance. Han *et al.* [5] proposed bidirectional LSTM to condition the compatible product based on previously selected products and get the compatibility score. In [4], type-aware embeddings are employed to understand the notion of compatibility between every pair of product categories. Authors in [8] have addressed both compatibility and its reasoning depending on attributes. Several works used image pairing with their similarity conditions to determine compatibility score, such as similarity condition embedding in [3], conditional similarity network in [9], and finding compatibility between each pair in an outfit in [10]. Similar to this approach, papers have used category-aware subspace [11] and attribute-level disentanglement [12] to get compatibility scores from embeddings created using an image, source, and target categories. Other lines of work include additional constraints for compatibility scoring, such as theme-based compatibility [13], context-based compatibility [14], scene-based compatibility [15], *etc*. The methods in this group ignore individual users' preferences and implicitly capture population preferences. Unlike these approaches, the proposed approach models user preference for compatibility prediction.

B. Personalized Outfit Compatibility

Works in this group model user preferences for compatibility prediction. Hu *et al.* [16] explored personalized recommendation problems using a functional tensor factorization model for user-item and item-item compatibility. FHN [2] associates different users with a trainable embedding to understand their preferences to facilitate personalized recommendations. PSA-Net [17] learns customer-dependent subspace learning method for personalized compatibility representation. LPAE [6] represents an outfit as a set and each user is associated with multiple learnable anchors for personalized compatibility modeling. Unlike these approaches, the proposed represents the outfit as a graph and uses GNN with dot attention to explicitly capture the relationships between fashion items. Further, we use an efficient approach for capturing user preferences for different styles.

III. Proposed method

In this section, the proposed approach for personalized outfit compatibility prediction is described in detail. Consider an outfit $O = \{I_1, I_2, .., I_N\}$ with N fashion items, $I_i \in R^{H \times W \times 3}$ i=1 to N represents the image of an item in the outfit, H and W represents the height and width of the image. Let u_l represents the l^{th} user in the system. Our aim is to predict the compatibility score ($s_O^{u_l}$) of outfit O for the user u_l.

Figure 1 depicts the block diagram of the proposed method. It consists of two streams, the top stream (visual stream) generates the outfit embedding, and the bottom stream models the user preferences. In summary, the visual feature extraction network (*e.g.*, ResNet-18 [18]) generates the visual features of the items. A fully connected graph is created to represent the outfit, where the nodes represent the items (components of the apparel). Each of these nodes is associated with its corresponding visual feature. This graph is updated using the dot attention GNN [19]. Finally, the graph read-out function generates the outfit embedding. We represent each user with a trainable vector (user embedding). The feature transformation network generates 'K' conditional embedding (style-conditioned user embedding) by transforming the user embedding. The compatibility score is generated by computing the cosine similarity between the outfit embedding and the style-conditioned user embedding. The details of each of these blocks are discussed in the following subsections.

A. Visual stream

(i) Visual feature extractor: The visual feature extractor module generates the visual embedding for each item (I_i) in the outfit (O) and is given by:

$$x_i = f_v(I_i; \theta_{f_v}) \tag{1}$$

Here, f_v represents the feature extractor with parameters θ_{f_v}, and $x_i \in \mathbb{R}^{d_{emb}}$ represents the visual embedding of dimension d_{emb}. We use ResNet-18 without the classifier layer for feature extraction. The output of global average pooling is fed to a FC (fully connected) layer to obtain the visual embedding of the item. We set the dimension of the embedding d_{emb}=256.

(ii) Dot Attention GNN: Post feature extraction, a graph $G_O = \{V, E\}$ is created, where V represents the set of items and E represents the set of edges. Visual embedding of the items forms the graph's node embedding. Let,

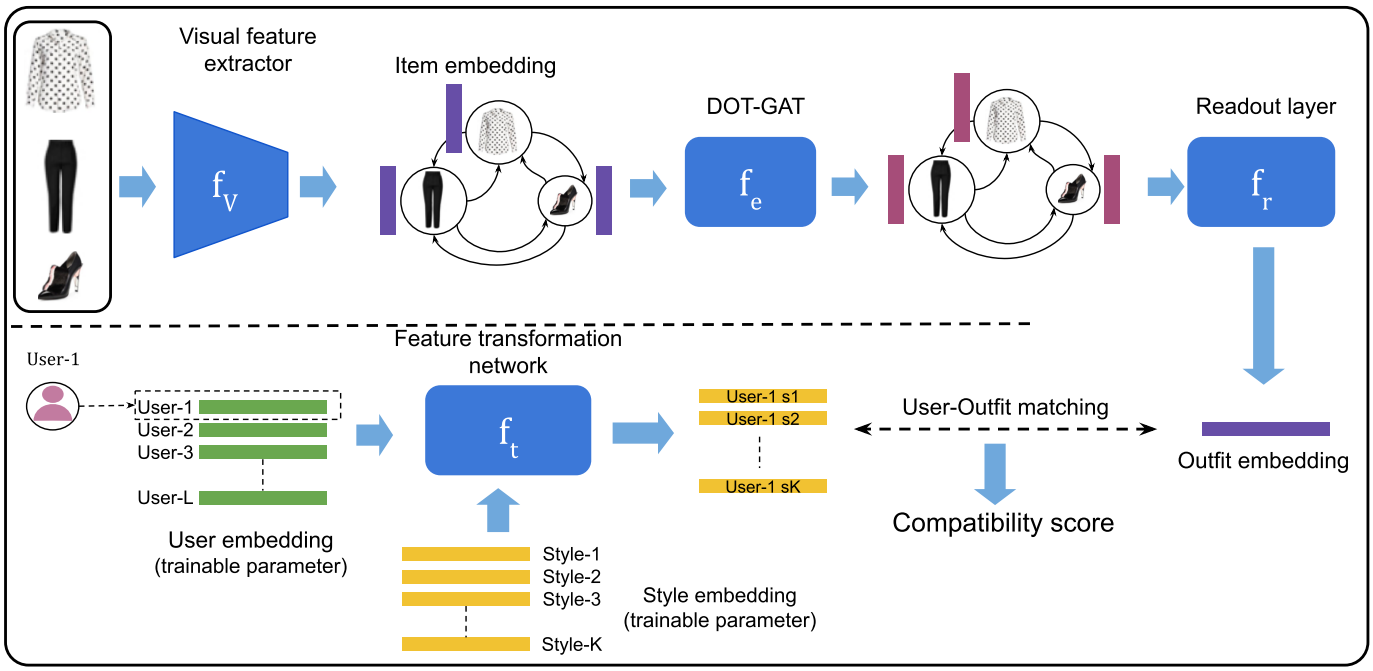

Fig. 1. Overview of the proposed approach. The top stream generates the outfit embedding, and the bottom stream models the user preferences. Top stream: feature extractor network extracts the visual features of the items, Dot-GNN updates the node embedding of the outfit graph, and the read-out layer generates the outfit embedding. Bottom stream: the feature transformation network generates the 'K' style-conditioned embedding for a user.

$X = [x_1; x_2; ....; x_N] \in \mathbb{R}^{N \times d_{emb}}$ represent the embedding of nodes. We use graph attention network [19] with dot attention mechanism to update the embeddings of the nodes. DOT-GNN uses a dot attention mechanism to capture visual interaction between the items. The updated node embedding is given by:

$$X^{new} = GNN_{DOT}(X, G_O; \theta_{gnn}) \tag{2}$$

where, $X^{new} = [x_1^{new}; x_2^{new}; ....; x_N^{new}] \in \mathbb{R}^{N \times d_{emb}}$ represents the updated embedding of nodes and θ_{gnn} represents the parameter of DOT-GAT. $\widehat{G}_o$ represents the graph with updated node embedding. The update equation for DOT-GNN is given by:

$$x_i^{new} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} x_j \tag{3}$$
$$\alpha_{ij} = softmax(e_{ij}, dim{=}1)$$
$$e_{ij} = (W_i x_i)^\top (W_j x_j)$$

Here, $\mathcal{N}_i$ represents the set of neighbor nodes of node i and θ_{gnn}=$\{W_i \in \mathbb{R}^{d_{emb} \times d_{emb}}, W_j \in \mathbb{R}^{d_{emb} \times d_{emb}}\}$ represents the trainable parameter. In graph convolutional network (GCN), equal importance is given to all the neighbor nodes ($j \in \mathcal{N}_i$) of a node (i), *i.e.,* α_{ij}=1. Unlike GCN, GNN with an attention mechanism gives different weightage ($0 \leq \alpha_{ij} \leq 1$) to neighbor nodes of a node. The attention helps to capture the asymmetrical importance of items in the outfit.

(iii) Outfit embedding generation: Post updating the graph's node embedding using DOT-GNN, the global representation of the graph is obtained *via* the graph read-out function. Read-out function is a permutation invariant function *e.g.*, mean, sum, max. The output of the read-out function, *i.e.,* outfit embedding is given by:

$$q = f_r(X^{new}, \widehat{G}_O) \tag{4}$$
$$f_r(X^{new}, \widehat{G}_O) = \frac{1}{N} \sum_{i=1}^{N} x_i^{new}; \textit{mean read-out} \tag{5}$$

Here, $q \in \mathbb{R}^{d_{emb}}$ represents the outfit embedding and the f_r represents the read-out function. Next, we explain the approach to model the user preferences.

B. Profiling stream

In Figure 1, the bottom stream captures the user's preferences. Typically, users have different preferences for different fashion styles. In the proposed method, each user is associated with an embedding ($u_l \in \mathbb{R}^{d_{emb}}$). To capture users' preferences for different styles or contexts, we transform the user embedding using the feature transformation network (f_t). The feature transformation block transforms the user embedding using 'K' learnable style embedding ($c_k \in \mathbb{R}^{d_{emb}}$), yielding 'K' style conditioned embedding ($u_{lk} \in \mathbb{R}^{d_{emb}}$) for each user.

$$u_{lk} = f_t(u_l, c_k; \theta_{f_t}); \textit{ for k=1 to K} \tag{6}$$
$$\textit{where, } f_t(u_l, c_k) = w^\top (u_l + c_k) + b$$

Here, θ_{f_t}=$\{w \in \mathbb{R}^{d_{emb} \times d_{emb}}, b \in \mathbb{R}^{d_{emb}}\}$ represents the trainable parameters. Both user (u_l) and style embedding (c_k) are randomly initialized. Then, with the aid of implicit feedback of the users on the outfit, these embedding are updated using bayesian personalized ranking (BPR) loss [20]. We set the dimension of user embedding (u_l) and style embedding (c_k)

equal to the dimension of outfit embedding (q), *i.e.,* d_{emb}=256. Further, we set the number of style embedding K=64.

C. Compatibility score

The compatibility score ($s_o^{u_l}$) of an outfit (o) to a user (u_l) is computed by taking the cosine similarity (CS) between outfit embedding (q) and style-conditioned user embedding (u_{lk}), given by:

$$s_o^{u_l} = \sum_{k=1}^{K} CS(u_{lk}, q) \tag{7}$$

Training: Let $\mathcal{O}_l^+ = \{o_1, o_2, ...\}$ represents the positive outfit set (outfit liked by user u_l) and $\mathcal{O}_l^-$ represents the negative outfit set. Consider mini-batch B with M samples, each sample represents a tuple (o_p,o_n,u_l) where $o_p \in \mathcal{O}_l^+$, and $o_n \in \mathcal{O}_l^-$ are positive and negative ouftit for user u_l. The proposed model is trained in an end-to-end manner. First, the compatibility scores for positive and negative outfits are generated. Then using bayesian personalized ranking loss [20], the entire network along with the user and style embeddings are updated. The training loss is given by:

$$\mathcal{L}_{total} = \frac{1}{M} \sum_{(o_p, o_n, u) \in B} log(1 + exp(-(s_{o_p}^u - s_{o_n}^u))) \tag{8}$$

Here, $s_{o_p}^u$ represents compatibility score of the positive outfit and $s_{o_n}^u$ represents compatibility score of the negative outfit.

IV. Experiments

In this section, we provide the quantitative and qualitative results to highlight the effectiveness of the proposed method. We follow the experimental setup of [2], and the details are as follows.

Dataset: we use Polyvore-U dataset [2] for the experiments. The details of the dataset are given in Table I. The dataset contains outfits collected by multiple users ('U'). Here, 'U' indicates the number of users. Polyvore-630 and 53 contain outfits of fixed length. Polyvore-519 and 32 contain outfits of variable lengths. Polyvore-53 and 32 datasets are used for evaluation in the cold start setting.

Metrics: For evaluation, we use the ranking metrics (i) Normalized Discounted Cumulative Gain (NDCG) and (ii) Area Under the ROC curve (AUC). The test set is ranked in descending order using the predicted compatibility score, and the ranking metrics AUC and NDCG are used for evaluation. Similar to [2], we consider two settings for evaluation, details given in Table II. During training, the ratio of positive to negative samples is 1:1. For evaluation, this ratio is set to 1:10. In protocol 1, negative outfits are randomly created by sampling fashion items of different categories (*i.e.,* top, bottom, and shoe). In protocol 2, outfits of other users are sampled to create negative samples for a given user.

TABLE I
DETAILS OF POLYVORE-U DATASET. POLYVORE-53 AND 32 ARE USED IN THE COLD-START SETTING.

Dataset	Users	Outfit size	Split	Items	Outfit
Polyvore-630	630	Fixed	Train	159729	127326
			Test	45505	23054
Polyvore-53	53		Train	20230	10712
			Test	4437	1944
Polyvore-519	519	Variable	Train	146475	83416
			Test	39085	14654
Polyvore-32	32		Train	14594	5133
			Test	2797	898

TABLE II
DETAILS OF TRAINING AND TESTING SETUP. HERE, THE RATIO REPRESENTS THE RATIO OF POSITIVE TO NEGATIVE SAMPLES. IN PROTOCOL 1, NEGATIVE OUTFITS ARE RANDOMLY CREATED BY SAMPLING FASHION ITEMS OF DIFFERENT CATEGORIES (I.E., TOP, BOTTOM, AND SHOE). IN PROTOCOL 2, OUTFITS OF OTHER USERS ARE SAMPLED TO CREATE NEGATIVE SAMPLES FOR A GIVEN USER.

	Train		Test	
	Negative outfit	Ratio	Negative outfit	Ratio
Protocol-1	Random online	1:1	Random fixed	1:10
Protocol-2	Hard online	1:1	Hard fixed	1:10

We use ImageNet [21] pre-trained ResNet-18 for feature extraction. The user and style embedding are randomly initialized. The proposed model is trained using an Adam optimizer [22] with a learning rate of 1e-4. The images are normalized with mean and standard deviation same as that used for pre-training of ResNet-18. For data augmentation, we perform random horizontal-flip during training. The training mini-batch size is set to 64.

We compare the proposed approach with the following methods:

- Bi-LSTM [5]: The method represents an outfit as a sequence of items. Each item is represented by its corresponding visual embedding. A bi-directional LSTM processes the sequence of visual embedding to predict the outfit compatibility score. The method does not consider user preference for compatibility prediction.
- Type-aware [4]: The method first extracts the embedding of the items in the general embedding space where semantically similar items will have similar embedding. Then using the learned projection, the item's general embedding is transformed into secondary embedding space for pair-wise item compatibility prediction. This method does not consider user preference for compatibility prediction.
- SCE-Net [3]: In this approach, the visual embedding of items is extracted using a CNN. Then, each pair of items are fed to a condition weight branch to determine the sub-space relevant for semantic comparison between item pairs. This method does not consider user preference for compatibility prediction.
- FHN [2]: In this method, the visual embedding of the

TABLE III
PROTOCOL-1 (RANDOM NEGATIVE OUTFITS). PERFORMANCE OF VARIOUS OUTFIT COMPATIBILITY PREDICTION METHODS. BI-LSTM, TYPE-AWARE, SCE-NET DOES NOT MODEL USER PREFERENCES. FHN (VISUAL), LPAE AND THE PROPOSED METHOD MODELS USERS' PREFERENCES.

Method	Polyvore-630		Polyvore-519	
	AUC	NDCG	AUC	NDCG
Bi-LSTM [5]	0.7840	0.6328	0.7882	0.6306
Type-Aware [4]	0.7727	0.6023	0.7840	0.6162
SCE-Net [3]	0.7966	0.6524	0.7849	0.6167
FHN [2]	0.8942	0.8090	0.8845	0.7784
LPAE-u [6]	0.9006	0.8186	0.9119	0.8359
Proposed	0.9107	0.8412	0.9237	0.8575

TABLE IV
PROTOCOL-2 (HARD NEGATIVE OUTFITS). PERFORMANCE OF VARIOUS OUTFIT COMPATIBILITY PREDICTION METHODS.

Method	Polyvore-630H		Polyvore-519H	
	AUC	NDCG	AUC	NDCG
Bi-LSTM [5]	0.4992	0.2817	0.4990	0.2739
Type-Aware [4]	0.5000	0.2790	0.4995	0.2740
FHN [2]	0.8132	0.6400	0.8129	0.6334
LPAE-u [6]	0.8314	0.6743	0.8385	0.6793
Proposed	0.8435	0.6932	0.8503	0.6970

fashion items is generated using a CNN. Then the item's visual embedding is fed to its corresponding category-specific type-embedding network to obtain the final item embedding. Users are associated with a trainable embedding to model their preferences. The outfit compatibility score is an aggregation of (i) item-item similarity score and (ii) user-item similarity score.

- LPAE [6]: This method generates the embedding of the items of an outfit using a CNN. Then the set of item embedding is processed by the set transformer [23], and the final outfit embedding is obtained by feeding the output of the set transformer to the attention-based pooling layer. The compatibility score is generated by matching user embedding and the outfit embedding.

Table III shows the performance of various outfit compatibility prediction methods on the Polyvore-630 and Polyvore-519 dataset. It can be observed that the performance of the proposed approach is better than the existing approaches. The results also highlight the need for modeling user preferences, *i.e.,* approaches that model users' preferences show superior performance compared to approaches that do not consider users' preferences (*i.e.,* Bi-LSTM [5], Type-aware [4], SCE-Net [3]). We attribute the superior performance of the proposed approach to (i) graph representation of the outfit and its processing using dot GNN; and (ii) use of style-conditioned user embeddings. Unlike FHE and LPAE, the proposed method explicitly captures the relationships between fashion items by representing the outfit as a graph. The dot GNN captures the asymmetric importance of the fashion items in the outfit graph.

Evaluation on the test set with hard negative outfit: Table IV shows the performance of the models on the test set with hard-negative samples. The proposed method shows superior performance compared to other approaches. Further, it can be observed that there is a drop in the performance of the models in the hard-negative evaluation setting compared to the random negative evaluation setting.

TABLE V
NEW USERS: PERFORMANCE OF VARIOUS OUTFIT COMPATIBILITY PREDICTION APPROACHES ON POLYVORE-53 AND 32 DATASET.

Method	Polyvore-53		Polyvore-32	
	AUC	NDCG	AUC	NDCG
FHN	0.8737	0.7763	0.8677	0.7449
LPAE-u	0.8822	0.7898	0.8954	0.8209
Proposed	0.8922	0.8062	0.9134	0.8477

TABLE VI
RESULT ON ABLATION EXPERIMENTS.

	Personlization	Style	Polyvore-630	
			AUC	NDCG
Ablation-1	No	No	0.7699	0.6803
Ablation-2	No	Yes	0.8035	0.7442
Proposed	Yes	Yes	0.9107	0.8412

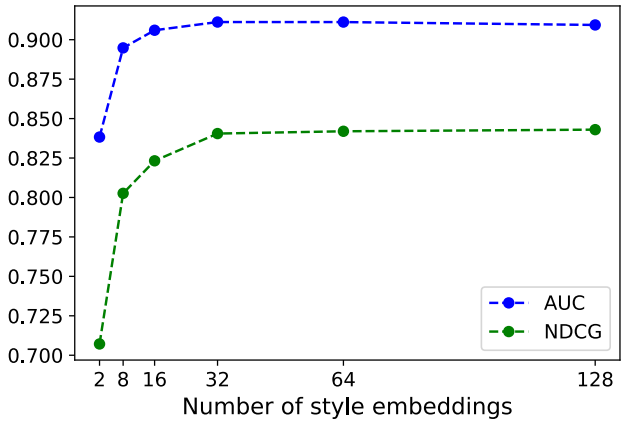

Fig. 2. Plot of AUC and NDCG metrics as a function of number of style embeddings used in the proposed model.

A. New users

In this subsection, we provide empirical results to highlight that the proposed approach can handle new users without re-training the entire network. It is common for an e-commerce platform to encounter new users. The scenario is referred to as a new user cold start problem where the RecSys has to cater to new users with little or no interaction data (purchase, viewed, add to cart). The outfit compatibility prediction module which is an integral part of the fashion outfit recommendation system should be able to handle new users. We consider the case where we have few samples for modeling user preferences. Training the entire model on new user data might cause overfitting of the network.

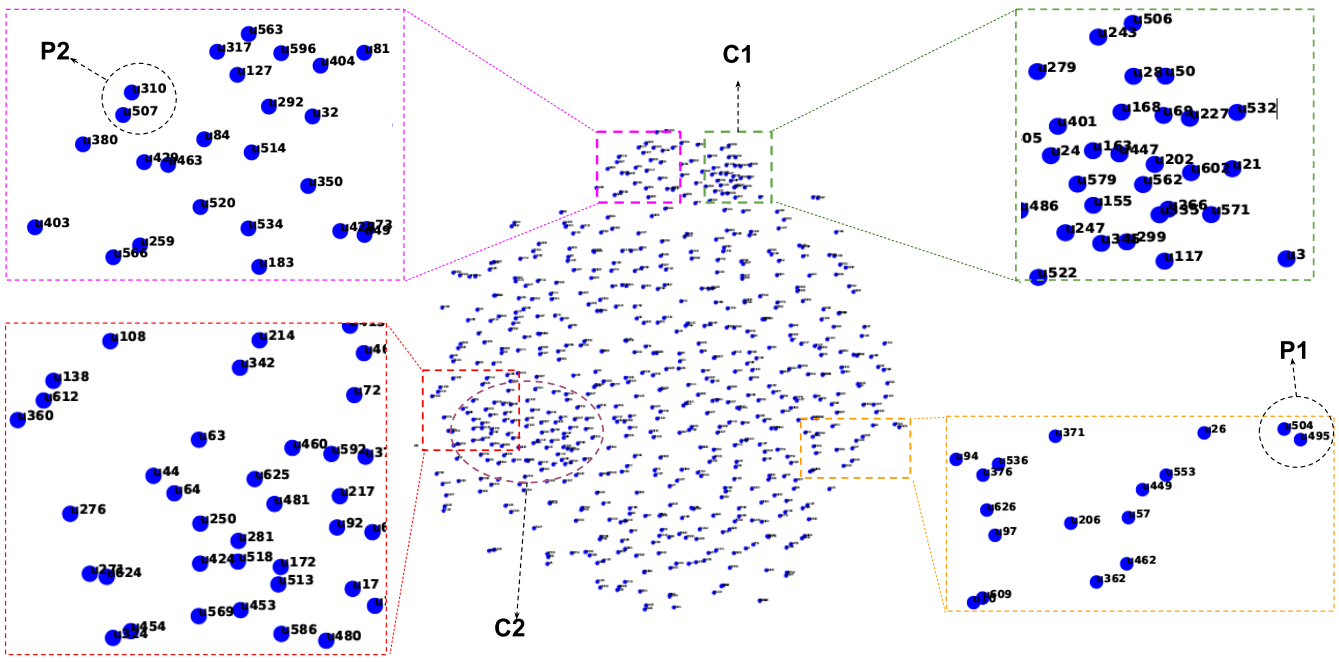

Fig. 3. t-SNE plot of user embedding. Here, we consider the user embedding of the proposed model trained on polyvore-630 dataset.

Fig. 4. Outfits and their compatibility scores for users 504, 495, 310, and 507. Note that, outfit compatibility scores for user pairs (504,495) and (310,507) are similar.

The proposed model uses an efficient and scalable approach for modeling user preferences. In the proposed approach, we freeze the network and optimize the user embedding of new users only. For Polyvore-53 and Polyvore-32 datasets, we consider the proposed models trained on Polyvore-630 and Polvyore-519, respectively. The weights of these models are frozen. Table V shows the performance of various methods in a cold-start setting. The performance of the proposed method is better than the existing methods. The results highlight the capability of the proposed method in handling new users. Furthermore, we can update the existing user embedding based on new interaction data without re-training the entire network.

B. Ablation

In this subsection, we provide experimental results to highlight the importance of various components of the proposed approach.

(i) Ablation-1: without personalization and style embedding. We train the proposed method without having user-specific embedding and style embedding. A single embedding is used to model population preferences. We perform this experiment

to highlight the need for modeling users' preferences. From Table VI, it can be observed that the performance of this model is inferior compared to the proposed model (personalization + style).

(ii) Ablation-2: without personalization and with style embedding. We trained the proposed method without having user-specific embedding. In this experiment, we use the feature transformation network to generate style-specific population embeddings. This experiment highlights the significance of style embeddings. Row 2 of Table VI shows the result for this experiment. It can be observed that there is an improvement in the performance of the model (without personalization) by adding style embedding (compare row-1 and row-2).

(iii) We train the proposed model with a different number of style embeddings. Figure 2 shows the NDCG and AUC of the model with a different number of style embeddings. It can be observed the AUC and NDCG increase with an increase in the number of style embedding and saturates for K$\geq$32.

In summary, the ablation experiments highlights the need for personalization (user embeddings) and the proposed feature transformation block that generates style-conditioned embeddings of a user.

C. Qualitative results

(i) t-SNE plot of user embedding: Figure 3 shows the t-SNE plot of the user embedding ($u_l \in \mathbb{R}^{d_{emb}}$). We consider the model trained on Polyvore-630 for generating this plot. Users with embedding nearby share similar preferences. The scatter pattern is due to the varying nature of user preferences. Further, we also observe the presence of communities, *i.e.,* a small group of users with similar preferences (C1 and C2 in Figure 3).

(ii) Outfit compatibility scores: Figure 4 shows the outfits and their compatibility score for different users. It can be observed that the outfit compatibility scores for users P1=(504,495) and P2=(310,507) are similar since users in P1 and P2 have nearby embedding (please refer to the Figure 3).

V. Conclusion

In this work, we consider the outfit compatibility prediction task, which is an important module in the fashion outfit recommendation system. This module helps in determining the visual compatibility of an outfit to a user. The variable outfit length, asymmetric importance of fashion items, and the presence of style-specific user preferences pose challenge for compatibility prediction. To address some of these challenges, we have proposed a novel approach for personalized outfit compatibility prediction. The proposed approach represents the outfit as a graph and uses dot-GNN to capture the relationship between fashion items. A graph read-out layer is used to generate the final outfit embedding. Style-specific user embeddings are generated using the proposed feature transformation network. The outfit compatibility score is generated by computing the similarity between the outfit embedding and the style-specific user embeddings. We have demonstrated the effectiveness of the proposed approach on the Polyvore-u dataset.

References

[1] "How retailers can keep up with consumers," https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers.

[2] Z. Lu, Y. Hu, Y. Jiang, Y. Chen, and B. Zeng, "Learning binary code for personalized fashion recommendation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 562–10 570.

[3] R. Tan, M. I. Vasileva, K. Saenko, and B. A. Plummer, "Learning similarity conditions without explicit supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 373–10 382.

[4] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 390–405.

[5] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1078–1086.

[6] Z. Lu, Y. Hu, Y. Chen, and B. Zeng, "Personalized outfit recommendation with learnable anchors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 722–12 731.

[7] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[8] X. Wang, B. Wu, and Y. Zhong, "Outfit compatibility prediction and diagnosis with multi-layered comparison network," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 329–337.

[9] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 830–838.

[10] M. Moosaei, Y. Lin, and H. Yang, "Fashion recommendation and compatibility prediction using relational network," *arXiv preprint arXiv:2005.06584*, 2020.

[11] Y.-L. Lin, S. Tran, and L. S. Davis, "Fashion outfit complementary item retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3311–3319.

[12] Y. Hou, E. Vig, M. Donoser, and L. Bazzani, "Learning attribute-driven disentangled representations for interactive fashion retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 147–12 157.

[13] J.-H. Lai, B. Wu, X. Wang, D. Zeng, T. Mei, and J. Liu, "Theme-matters: Fashion compatibility learning via theme attention," *arXiv preprint arXiv:1912.06227*, 2019.

[14] G. Cucurull, P. Taslakian, and D. Vazquez, "Context-aware visual compatibility prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 617–12 626.

[15] W.-C. Kang, E. Kim, J. Leskovec, C. Rosenberg, and J. McAuley, "Complete the look: Scene-based complementary product recommendation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 532–10 541.

[16] Y. Hu, X. Yi, and L. S. Davis, "Collaborative fashion recommendation: A functional tensor factorization approach," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 129–138.

[17] M. Taraviya, A. Beniwal, Y.-L. Lin, and L. Davis, "Psanet-subspace attention for personalized compatibility," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1354–1360.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.
[20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.
[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[23] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3744–3753.