2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897234

LEARNING FROM DESIGNERS: FASHION COMPATIBILITY ANALYSIS VIA DATASET DISTILLATION

Yulan Chen *Zhiyong Wu* *Zheyan Shen* *Jia Jia*⋆

Department of Computer Science and Technology, Tsinghua University
Beijing National Research Center for Information Science and Technology

ABSTRACT

Learning fashion compatibility is of great significance to both academic research and industry, which serves as a key technique for many real applications like online shopping recommendation and clothing generation. In previous studies, user-generated data (e.g. outfits from social media platform) are usually used for learning item embeddings and further modeling the compatibility. However, due to the noisy and messy nature of such data, one can hardly learn a representation that can clearly characterize the fashion-related attributes (e.g. color, material). In this paper, we propose an Attention-based Dataset Distillation Graph Neural Network (ADD-GNN) to leverage the designer-generated data as a guidance on modeling the outfit compatibility. Specifically, we jointly optimize two components which distill knowledge from fashion designers for feature representation learning and model the overall compatibility through attention-based graph neural network. Experimental results on real world fashion datasets clearly demonstrate the superiority of our proposed ADD-GNN against several competitive baselines in outfit compatibility tasks, which proves the effectiveness of distilling knowledge from designers.

Index Terms— fashion compatibility, dataset distillation, graph neural network

1. INTRODUCTION

Fashion plays an important and complex role in our social life. Fashion style can be considered as common knowledge of people about a certain type of clothing and the extended meaning it represents. Therefore, learning fashion compatibility has become an attractive research direction in recent years[1, 2], which can be further applied to online shopping recommendation and clothing generation.

Nowadays people are used to learn fashion trends from both fashion designers and other users from the Internet. Many famous fashion brands present their new designs on fashion shows, which may lead the fashion trend over a year. People also share and comment their outfits on social network and online shopping sites. In a word, it is more convenient for people to enhance their fashion sense through daily social life.

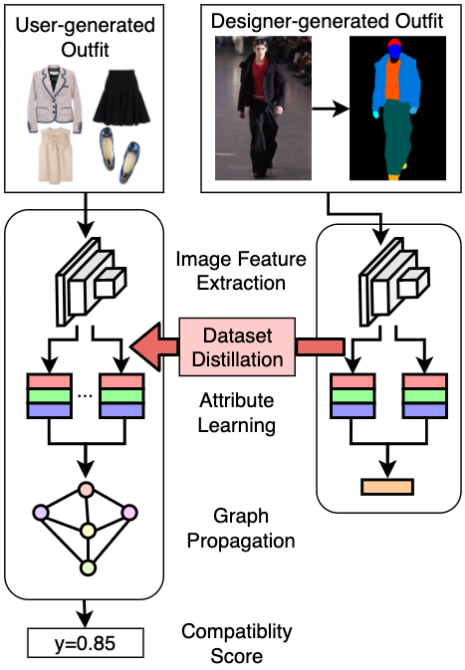

Fig. 1. The framework of our proposed model.

However, it is not the case in the fashion research field. By so far, there are few researches considering fashion knowledge from designer-generated and user-generated data. In previous studies, researchers usually leverage user-generated data to extract the embeddings of the fashion items and learn the compatibility scores of the outfits, either through a pairwise scheme [3, 4] or by considering the compatibility of the whole outfit [5, 6, 7]. Due to the noisy and messy nature of user-generated data on the social media platform, it is usually difficult for traditional methods to acquire high quality item embeddings which can clearly characterize the complex fashion-related attributes (e.g. color, material), causing the

⋆Corresponding author : Jia Jia.

This work was supported by the National Key RD Program of China under Grant No.2021QY1500, the state key program of the National Natural Science Foundation of China (NSFC) (No.61831022).

unsatisfactory performance on predicting outfit compatibility. This leads us to the question: how to leverage the expert knowledge from fashion designers to guide the modeling of complex correlation between clothing attributes, and further exploit it to better study the overall compatibility of outfits.

To study the development of fashion styles, [8] has introduced a large fashion show dataset, which contains images of outfits from different fashion designers and serves as an ideal external source of fashion compatibility learning task. Moreover, we are enlightened by the dataset distillation techniques introduced by [9], as an alternative formulation of traditional knowledge distillation method [10]. Instead of distilling knowledge of a complex model into a simpler one, dataset distillation aims to distill knowledge of one dataset into another one. Inspired by theses works, we try to leverage the knowledge from designers and apply it to our compatibility learning problem through a dataset distillation scheme.

In this paper, we propose an Attention-based Dataset Distillation Graph Neural Network (ADD-GNN) which synergistically distill knowledge from fashion designers to guide the study of fashion attributes as well as capture the hidden correlations between clothing items through an attention-based graph neural network. The framework of our model is shown in Fig 1. Specifically, we consider the whole set of fashion items as a fashion graph, where the categories of the fashion items are the nodes. Each outfit is made up of a set of fashion items, which appears to be its subgraph. We first extract image features of the clothing items using convolutional neural network. We learn several fashion attributes from the image features and get the importance of different attributes using attention mechanism. We align the latent features learned from the teacher network with those from student network. Then we get the aggregated embedding features of each node using a graph neural network. Finally we add the pair-wise compatibility scores of the items, and get the compatibility score of the whole outfit. Empirical results on real-world datasets demonstrate the effectiveness of our proposed method in terms of AUC and FITB accuracy.

Our contributions are described as follows:

- We are the first to introduce expert knowledge from fashion designers into fashion compatibility analysis.
- We propose a synergistically model to both learn the hidden correlation of fashion attributes and describe the relationship between clothing items.
- We conduct experiments on user-generated dataset and improve the compatibility score by introducing knowledge from designer-generated dataset.

2. RELATED WORKS

Previous works have been done on pair-level compatibility. [3] learns type-aware embeddings in subspaces that can capture similarities in different attributes. [4] proposes a content-based neural scheme to model fashion compatibility. These researches are difficult to capture the complex relationship between items. It is also easy to omit the contribution of accessories to the overall style of clothing.

Later works take into account outfit-level compatibility. [5] proposes a bidirectional LSTM (Bi-LSTM) model which sequentially predict the next item of the outfit. [7] employs multiplication of an individual feature with the gradient of the output score to quantify the influence score of each item feature. [11] first introduces graph convolution network in outfit compatibility prediction. These works learn fashion compatibility through massive online data provided by users. The information was messy and could not accurately reflect the fashion trend.

3. METHODOLOGY

3.1. Problem formulation

Problem Given a set of fashion outfits $O = (o_1, o_2, o_3, \dots)$ and a fashion graph G, our goal is to measure the compatibility score y_o of an outfit $o \in O$.

Definition 1. Each outfit $o = (x_1, x_2, x_3, \dots)$ contains $|o|$ fashion items. Each fashion item x_i belongs to a fashion category c_i, and is made up of K attributes.

Definition 2. The fashion graph $G = (N, E, A)$ is a directed graph, where each node $c_i \in N$ represents a clothing category, and $A \in \mathbb{R}^{|N|\times|N|}$ is the weight matrix of the graph. $A_{ij} \in A$ represents the frequency of two categories c_i and c_j appearing together in the whole outfit set. For each outfit o, the fashion items in o form a subgraph G_o.

3.2. Attribute learning

We suppose there are K latent attributes in one particular clothing item. These attributes are extracted by K masks, and then their features are concatenated, resulting in the following feature representation:

$$m = (\sigma(M^0 f), \dots, \sigma(M^K f)), \tag{1}$$

where f means the original feature extracted from the image by convolutional neural networks. m means the masked feature. $M \in \mathbb{R}^{|f|\times K}$ are the learned masks.

To get disentangled masks, we add a loss L_M:

$$L_M = ||M||_1 = \max_j \sum_{k=1}^{K} |M_j^k|, \tag{2}$$

where $|M_j^k|$ means the absolute value of M_j^k.

3.3. The graph neural network

The graph neural network is first introduced by [12], which has been widely used to model graph-structured data. In our

model, the graph neural network is used to aggregate the item features in the outfit and get a final item representation.

Before training, we calculate the initial edge weights as follows:

$$A_{ij} = \frac{count(c_i, c_j) + \lambda}{count(c_i) + |N| \cdot \lambda}, \tag{3}$$

where λ is a smoothing parameter.

We get the initial node feature $m_i = (m_{i0}, \ldots, m_{iK})$ of item image feature f_i through masking, as is mentioned above.

Then we can get the initial hidden state $h_i^0 = (h_{i0}^0, \ldots, h_{iK}^0)$ of the node by the following equation:

$$h_{ik}^0 = \sigma(W_0 m_{ik} + b_0), \tag{4}$$

where σ is the activate function, W_0 is the weight matrix and b_0 is the bias.

When modeling one outfit, we extract a subgraph from the original fashion graph by selecting all categories that belong to the items. To keep the features stable, we normalize the weights of a certain node.

In the graph neural network, we get the aggregated node representations. At the aggregation step t, we have received the last hidden state h_{ik}^{t-1}. Then we calculate a_{ik}^t as the weighted sum of all hidden states in the outfit:

$$a_i^t = \sum_{c_j \in G_o} A_{ij} h_j^{t-1}. \tag{5}$$

Then the hidden state h_{ik}^t with the hidden size $|h|$ is updated as:

$$\begin{aligned} z_t &= \sigma(W_z a_{ik}^t + U_z h_i^{t-1}), \quad r_t = \sigma(W_t a_{ik}^t + U_t h_{ik}^{t-1}), \\ \tilde{h_t} &= tanh(W_h a_{ik}^t + U(r_t \odot h_{ik}^{t-1})), \\ h_{ik}^t &= (1 - z_t) \odot h_{ik}^{t-1} + z_t \odot \tilde{h_t}, \end{aligned} \tag{6}$$

where W_z,W_t,W_h are the corresponding weight matrices.

After aggregation, we have the final node representation $h_i^t = (h_{i0}^t, \ldots, h_{iK}^t)$ of node c_i. For each item pair h_i^t and h_j^t, the pair-wise compatibility score is:

$$P(i, j) = \sum_{k=1}^{K} h_{ik}^t W_k h_{jk}^t, \tag{7}$$

where $W \in \mathbb{R}^{K \times |h| \times |h|}$ means the attention weight of K attributes.

We calculate the outfit compatibility score y as the average of the weighted sum of the pair-wise similarity:

$$y = \frac{1}{|G_o|(|G_o| - 1)} \sum_{c_i, c_j \in G_o} P(i, j). \tag{8}$$

We consider an outfit from the dataset as positive outfit x_{pos}. For each positive outfit, we randomly substitute one item with a randomly chosen item from the whole dataset to form a negative sample x_{neg}. y_{pos} and y_{neg} are their compatibility scores. The loss of the final output L_O is:

$$L_O = \sum_{(y_{pos}, y_{neg}) \in O} -ln\sigma(y_{pos} - y_{neg}), \tag{9}$$

where σ is the activate function.

3.4. Knowledge distillation

We train our teacher network on the fashion show dataset, which will be introduced in detail in section 4. In the fashion show dataset, their are only two parts (top and bottom) in one outfit. So we do not apply the graph neural network.

For each outfit $o = (x_0, x_1)$, the original feature of x_i is f_i. Then we can get the compatibility score as follows:

$$y = P(0, 1) = \sum_{k=1}^{K} m_0^k W_k m_1^k, \tag{10}$$

where m_i is calculated by Eq 1.

The loss between the teacher network and the student network is:

$$L_D = \sum_{k=1}^{K} ||m_{student}^k - m_{teacher}^k||_2. \tag{11}$$

3.5. Optimization

We formulate the objective function as:

$$L = L_O + \lambda_1 L_M + \lambda_2 L_D + \lambda_3 ||\Theta||, \tag{12}$$

where Θ refers to the set of parameters, λ_1, λ_2 and λ_3 refer to the hyper-parameters.

4. EXPERIMENTS

4.1. Datasets

We conduct our experiments on the following datasets:

(1) Fashion32 Dataset[13]: It is built up with fashion images from an e-commerce website[1], which represents user-generated data. It contains 13,904 outfits in which the fashion items belong to 90 categories[2].

(2) Fashion Show Dataset[8]: It is built up with fashion show images from Vogue[3], which represents designer-generated data. It contains 550 fashion brands in 10 years[4]. We select 30,058 outfits with both top and bottom clothes.

The datasets are split into 3 parts: training set, validation set and test set. The proportion of each part is 7:1:2.

[1] JD.com

[2] http://www.larry-lai.com/fashion.html

[3] vogue.com

[4] https://pan.baidu.com/s/1boPm2OB

4.2. Experimental setup

4.2.1. Evaluation tasks

To verify the performance of the proposed model, we evaluate it on the following two tasks:

(1) Fill-in-the-blank(FITB): Given a set of fashion items from one outfit, our goal is to select one item from multiple choices that is compatible with other items.

(2) Compatibility prediction: We use the area under the receiver operating characteristic curve (AUC) as the metric.

4.2.2. Experimental settings

We build a feature extractor using Resnet50 [14], which is pre-trained on ImageNet [15]. We remove the last fully connected layer. For each fashion image, we extract 2048-dimensional feature data.

The number of attributes K is 4. The number of hidden units of the graph is 256. The aggregation step in GNN is 2. We use RMSprop as the optimizer and set the parameter $\lambda_1 = \lambda_2 = \lambda_3 = 0.001$. The learning rate is 0.001.

4.2.3. Comparison methods

We compare our model with several prior works:

Bi-LSTM[5] This work is the first outfit-level fashion compatibility model. It sequentially predicts the next item conditioned on previous ones to learn their compatibility. We only use the visual part of the model.

NGNN[11] This is the first model to represent outfit as a graph. We train the visual model.

4.3. Experimental results

4.3.1. Performance analysis

Model	AUC	FITB Accuracy
Bi-LSTM	0.849	0.604
NGNN	0.862	0.691
ADD-GNN	**0.883**	**0.731**

Table 1. Experimental results on Fashion32 dataset, higher score indicates better performance.

We conduct experiments on both fill-in-the-blank and outfit compatibility tasks. The results are shown in Table 1. All methods have higher AUC score because it is simpler for all methods to identify the more compatible outfit from two choices than from multiple choices.

The experimental results confirm the effectiveness of our model. On this basis, we further analyze the following aspects:

1) Compared with sequence model, graph models have higher AUC and FITB scores, which proves the effectiveness of considering the mutual interaction between clothing items.

2) Compared with NGNN, our model gets a 2.1% and a 4.0% improvement on AUC and FITB accuracy. The results prove that using dataset distillation and focusing on disentangled attributes can benefit fashion compatibility prediction.

4.3.2. Ablation Studies

Model	AUC	FITB Accuracy
ADD-GNN w.o. Knowledge	0.860	0.688
ADD-GNN w.o. Graph	0.867	0.724
ADD-GNN	**0.883**	**0.731**

Table 2. Performance analysis of different model variants. Here "w.o. Knowledge/Graph" means we exclude the data distillation/GNN component from the model.

Table 2 shows the contributions of different components in our model. When adding dataset distillation, we get a 2.3% and a 4.3% improvement on AUC and FITB accuracy. When using the graph structure, we get a 1.6% and a 0.7% improvement on AUC and FITB accuracy.

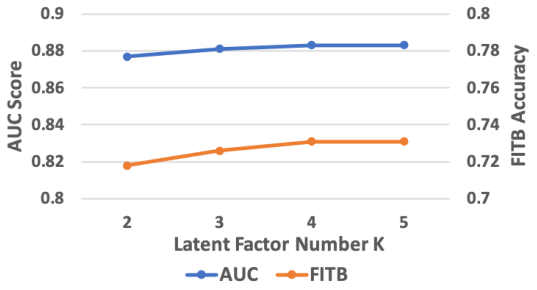

Fig. 2. Performance analysis of different attribute number K.

To explore the performance of attribute learning, we compare the AUC and FITB accuracy of different attribute number K. Figure 2 shows the results. When the attribute number K increases, AUC and FITB accuracy increases at first and then stay stable, which means the separation of attributes can benefit compatibility learning.

5. CONCLUSIONS

In this paper, we propose an Attention-based Knowledge Distillation Graph Neural Network to model fashion compatibility. We learn the importance of different attributes in fashion items from fashion designers. We model the outfit using graph structure, which can learn the relationship between clothing items. The experimental results prove the effectiveness of our method. Our approach can be applied to many areas, including online fashion recommendation. In the future, we can adopt more personal information into fashion modeling, such as user preference.

6. REFERENCES

[1] Reeta Koshy, Anisha Gharat, Tejashri Wagh, and Siddesh Sonawane, "A complexion based outfit color recommender using neural networks," in *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. IEEE, 2021, pp. 1–7.

[2] Na Zheng, Xuemeng Song, Qingying Niu, Xue Dong, Yibing Zhan, and Liqiang Nie, "Collocation and try-on network: Whether an outfit is compatible," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 309–317.

[3] Andreas Veit, Serge Belongie, and Theofanis Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 830–838.

[4] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma, "Neurostylist: Neural compatibility modeling for clothing matching," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 753–761.

[5] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis, "Learning fashion compatibility with bidirectional lstms," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1078–1086.

[6] Xun Yang, Yunshan Ma, Lizi Liao, Meng Wang, and Tat-Seng Chua, "Transnfcm: Translation-based neural fashion compatibility modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 403–410.

[7] Pongsate Tangseng and Takayuki Okatani, "Toward explainable fashion recommendation," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2153–2162.

[8] Yihui Ma, Jia Jia, Suping Zhou, Jingtian Fu, Yejun Liu, and Zijian Tong, "Towards better understanding the clothing fashion styles: A multimodal deep learning approach," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros, "Dataset distillation," *arXiv preprint arXiv:1811.10959*, 2018.

[10] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[11] Zeyu Cui, Zekun Li, Shu Wu, Xiao-Yu Zhang, and Liang Wang, "Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks," in *The World Wide Web Conference*, 2019, pp. 307–317.

[12] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," 2016.

[13] Jui-Hsin Lai, Bo Wu, Jingen Liu, Xin Wang, Dan Zeng, and Tao Mei, "Theme-matters: Fashion compatibility learning via theme attention," *arXiv preprint arXiv:1912.06227*, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.