Dress Well via Fashion Cognitive Learning

Kaicheng Pang[1,2]
kaicpang.pang@connect.polyu.hk
Xingxing Zou[1,2]
aemika.zou@connect.polyu.hk
Waikeung Wong†[1,2]
calvin.wong@polyu.edu.hk

[1] School of Fashion and Textiles
The Hong Kong Polytechnic University
Hong Kong SAR

[2] Laboratory for Artificial Intelligence in Design
Hong Kong SAR

Abstract

Fashion compatibility models enable online retailers to easily obtain a large number of outfit compositions with good quality. However, effective fashion recommendation demands precise service for each customer with a deeper cognition of fashion. In this paper, we conduct the first study on fashion cognitive learning, which is fashion recommendations conditioned on personal physical information. To this end, we propose a Fashion Convolutional Network (FCN) to learn the relationships among visual-semantic embedding of outfit composition and appearance features of individuals. FCN contains two submodules, namely outfit encoder and Multi-label Graph Convolutional Networks (ML-GCN). The outfit encoder uses a convolutional layer to encode an outfit into an outfit embedding. The latter module learns classifiers for physical labels via stacked GCN. We build a new dataset named Outfit for You (O4U) that contains 29,352 valid outfits with 5.25 unmatched physical labels on average. Extensive experiments are conducted on the O4U dataset and the quantitative results on O4U show that our proposed approach outperforms alternative approaches by clear margins. All data can be found at https://github.com/AemikaChow/AiDLab-fAshIon-Data.

1 Introduction

Fashion exists [14] in our daily life as a tool for expressing attitude and presenting culture. However, there exists a problem when applying it to online products since the fashion compatibility models only solved the task that fashion items in an outfit are compatible with each other but have not considered whether the outfits are compatible with customers when they are shopping online. As shown in Figure 1 (b), different customers have varied appearances, *e.g.*, different heights, hairstyles, skin colors, *etc.*, which will directly affect whether an outfit is compatible with them or not. For example, the outfit consists of a white long dress that is not suitable for the second customer since she is not so high enough to wear this long dress. Thus, even though this outfit itself is perfectly matched, it is inappropriate to recommend this outfit to her. Otherwise, it may be resulting she losing trust in the service provider. In other words, understanding relationships between outfits and customers to achieve precise outfit recommendations is crucial.

† The corresponding author.

(a) Outfit compositions generated by fashion compatibility models

(b) Precise Outfit Recommendation with fashion cognitive model

Figure 1: Current situations occur in fashion online retailing. Even though the outfit in the second row is perfectly matched, it is inappropriate to recommend this outfit to the second customer since she is not so high to match the white long dress.

Previous research mainly focused on the relations among fashion items via fashion compatibility learning [4, 10, 25, 27]. Many of them [10, 30] also focused on the explainability of fashion compatibility models. In addition, a few works noticed the influence of personal information, *e.g*. user preference [2, 12, 15], social media posts [21, 28], body shape [6]. However, no prior approach systematically considered the compatibility relationships between fashion items in an outfit and the varied appearance of online shoppers.

In this work, we aim to provide precise and appropriate fashion recommendation service to customers by considering their personal physical information. To distinguish from previous works utilizing the user's personal preference for personalized recommendation, we define the new task as **Fashion Cognitive Learning**, *i.e*., focusing on the influence of personal physical information on the compatibility of an outfit. To achieve this, we treat this task as a multi-label classification task and propose an end-to-end framework, namely Fashion Convolutional Network (FCN), that learns the compatible relationships between outfits and humans. The FCN contains two modules, namely outfit encoder and Multi-label Graph Convolutional Networks (ML-GCN). The outfit encoder utilizes several convolutional filters with different window sizes to encode the outfit into an outfit embedding. Applying filters with different sizes enables convolutional kernels to see different combinations of fashion attribute features. The ML-GCN is employed to learn multi-label classifiers based on word embeddings of physical labels. The predicted scores for all labels are obtained by multiplying classifier vectors with outfit embeddings.

Meanwhile, to facilitate the development of our framework, we introduce a new outfit dataset covering personal physical information, namely Outfits for You (O4U). The O4U focuses on women's wear since women are the largest market among all types of crowd [1] and all labels are designed according to women's characteristics. It includes a total of 29,352 outfits. Each outfit has two types of the label: 1. this outfit is good or not; 2. this outfit is not compatible with which kind of physical label. We invited six fashion experts to label these outfits and the labeling procedure is carefully designed to maintain annotation consistency. Extensive experiments are conducted on the O4U dataset and the results show that

our proposed FCN outperforms other baselines. The main contributions of this work are summarized as follows:

- We introduce a new task, i.e., fashion cognitive learning, which targets learning the compatibility relationships between outfits and personal physical information in an end-to-end framework to facilitate precise fashion recommendations.
- We introduce a new outfit dataset with tremendous personal physical information for facilitating fashion cognitive learning.
- Through extensive experiments, we demonstrate our network outperforms several alternative methods with clear margins.

2 Related Work

2.1 Personal Fashion Recommendation

Personalization is vital to all online selling services [1]. Some work focused on recommending items based on the user preference [2, 9, 12, 13, 15, 19], *e.g*., purchasing records, social media posts [21, 28], *e.g*., information from Instagram, or body shape [6]. Specifically, Packer *et al*. proposed an approach to personalizing clothing recommendations that models the dynamics of individual users' visual preferences by using interpretable image representations generated with a unique feature learning process [15]. Wen *et al*. constructed knowledge graphs of the user, clothing, and context, and utilized the Apriori algorithm to capture the intrinsic correlations between clothing attributes and context attributes [26]. Chen *et al*. connected user preferences regarding individual items and outfits with Transformer architecture [2]. Zheng *et al*. presented an item-to-set metric learning framework that learns to compute the similarity between a set of historical fashion items of a user to a new fashion item [28]. Kim *et al*. proposed a KD framework for outfit recommendation, which exploits false-negative information from the teacher model while not requiring the ranking of all candidates [7]. Different from the above personal fashion recommendation, this work defines a new task, *i.e*., Fashion Cognitive Learning, which learns the compatibility between outfits and varied personal physical information. Hidayati [6] learned the compatibility of clothing styles and body shapes which belongs to the scope of this new task. However, they only focused on one aspect, *i.e*., body shape. In addition, the way they constructed the dataset was to crawl images of stylish female celebrities by issuing each stylish celebrity name combined with a clothing category to be collected as a search query. We follow the common practice of fashion compatibility learning to build the O4U dataset with varied labels of personal features and the procedure of building the O4U dataset is presented in the next section.

3 O4U Dataset

Fashion cognitive learning is based on fashion compatibility learning to further learn the compatibility between outfits and personal physical features. Thus, we build the Outfit for You (O4U) dataset following the same structure of fashion compatibility learning.

Firstly, the label system, *i.e*., which types of personal physical information may influence the compatibility with outfits, is designed by professors from top-tier fashion design schools and recognized by our collaborated experts from fashion retailing groups. The details of the

Table 1: Details of personal physical features and their sub-features.

Features	**Sub-features** (N - numbers of sub-features)
Body Shape	rectangle, top hourglass, athletics, diamond, round, spoon, bottom hourglass... (10)
Skin Color	yellow, dark, fair, brown (4)
Hair Style	long curls, long straight hair... (6)
Hair Color	ginger,black, dark brown, light brown... (6)
Height	high, middle, low (3)
Breasts Size	big, average, small (3)
Color-contrast	high, low (2)

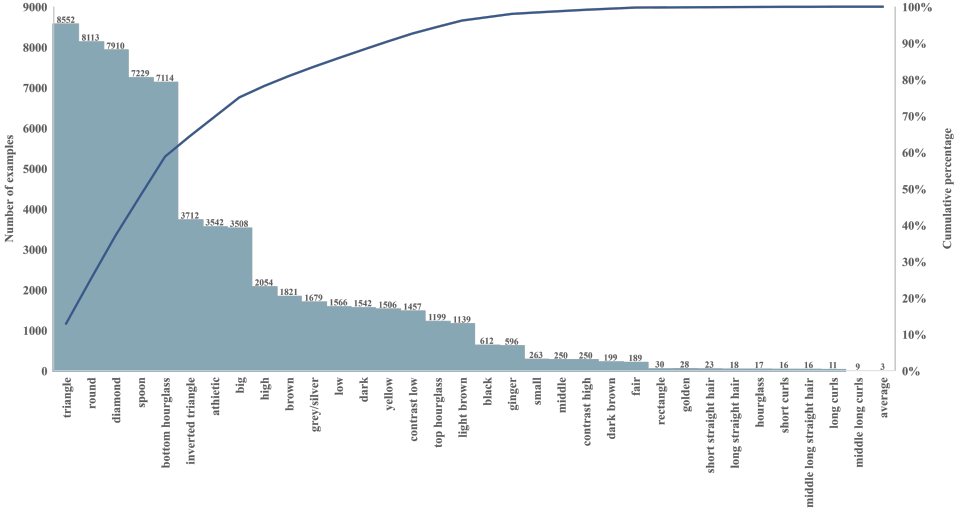

Figure 2: Number of examples for each physical label

defined label system are summarized in Table 1. Then, to ensure the objectiveness of the created outfits to the maximum extent, we randomly generate 50,000 seed outfits consisting of at least clothing items covering the whole body, one bag, one pair of shoes, and n accessories ($n \in [0,5]$). We invite six experts majoring in fashion to label those outfits. Determining if an outfit is well-matched is the first step. If true, they will further select which personal features are not compatible with this outfit. Otherwise, this outfit only has a label to indicate that it is not well-matched. An outfit is only kept if the consistent accuracy of these six experts is over 95% in all 34 labels. The few inconsistent annotation results are decided by the voting mechanism. After the labeling process, there are 29,352 outfits are retained. Meanwhile, only 15,748 outfits are labeled as well-matched and the average unmatched physical label of these well-matched outfits is 5.25. We randomly divide the dataset into a training set, validation set, and test set in the form of 8:1:1. The label distribution of the training set is shown in Figure 2.

4 Approach

4.1 Problem Formulation and Motivation

We define a new task, w.r.t. Fashion Cognitive Learning, which aims to learn the compatibility between outfits and personal physical information. Specifically, given a set of items

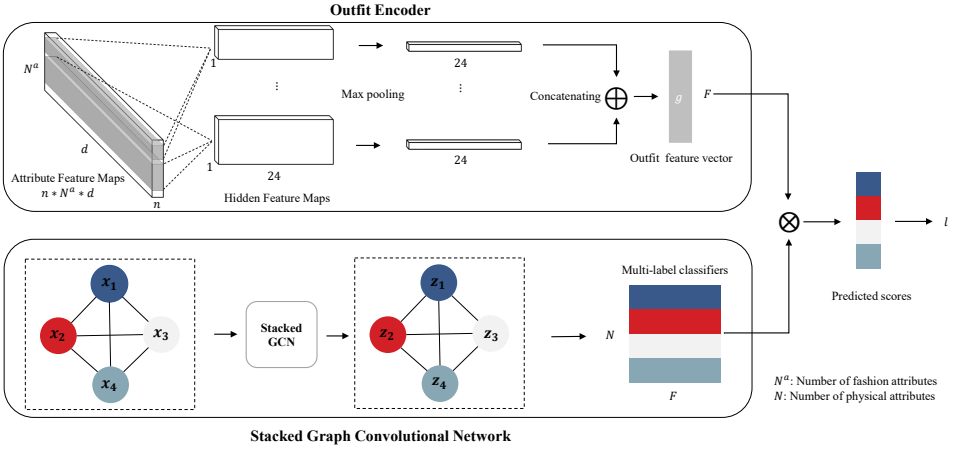

Figure 3: An overview of the proposed Fashion Convolutional Network. It comprises an outfit encoder and a stacked graph convolutional networks. We utilize the outfit encoder to encode outfits into outfit feature vectors by applying convolutional operation on attribute features. At the bottom of this figure, we exploit a stacked graph convolutional networks to represent the classifiers of physical labels. Each physical label is treated as a node of the graph. The predicted scores are obtained by applying these label classifiers to the outfit feature vector.

$\mathcal{M} = \{p_i\}_i^{N_p}$ of N_p individual items and a collection $\mathcal{T} = \{O_j\}_{j=1}^{N_t}$ containing N_t outfits, each outfit $O = \{p_i\}_i^n$ in collection $\mathcal{T}$ is defined as a subset of $\mathcal{M}$ containing n different items. Each outfit O has a fashion compatibility label $l_f \in \{0,1\}$ indicating whether this outfit is well-matched or not and a set of personal physical labels $\mathbf{l_p} \in \mathbb{R}^N$, where N is the number of physical labels defined in Table 1. Each item $p_i \in \mathcal{M}$ has its corresponding image I_i (unstructured data) and other metadata such as the primary color data, the category label, and some attribute labels $\mathbf{l_a}$ (structured data). Fashion cognitive learning is defined as a multi-label classification task that aims to recognize whether the given outfit is compatible with personal physical labels.

How to encode an outfit into a meaningful embedding is crucial for fashion cognitive learning. In this work, we propose to use 1-dimensional convolutional filters of different sizes to extract the hidden features of the outfit. The motivation for using a convolutional structure to encode outfits is mainly twofold: 1. we observed that translation equivalence exists in outfit data, *i.e.*, if we swap the order of items or attributes, the outfit embedding should remain the same. Data with such characteristics is suitable to be learned through a convolutional model; 2. we observe that whether an outfit is compatible with a physical label depends on one or several fashion attributes. Thus we use convolutional filters with different sizes to group different numbers of attribute features together.

4.2 Fashion Convolutional Network

The proposed Fashion Convolutional Network (FCN) contains two modules namely outfit encoder and stacked graph convolutional networks.

4.2.1 Outfit Encoder

To better reveal the continuity of outfit data, we propose to use a convolutional network to encode an outfit into an outfit embedding, as shown in Figure 3. Different from using a convolutional neural network (CNN) to extract features from item images, our proposed outfit encoder is applied to fashion attribute features. Given an outfit, fashion attribute features $\mathbf{X} \in \mathbb{R}^{N_a \times d}$ is extracted from each item using well-pretrained VGG [20] network on a large-scale fashion attribute dataset, where N_a is the number of fashion attributes and d is the dimensionality of attribute features. Each attribute feature vector is the output of the last convolution layer after a max-pooling operation. These attribute features, serving as the input of the outfit encoder, are fixed during the whole training process. The advantage of using these attribute features compared with using raw images is that the network can focus on the important features of items and make the training process more efficient.

Attribute feature maps of the given outfit with n items are presented by stacking (padded where necessary) all attribute features along the item dimension,

$$\mathbf{Z} = \mathbf{X}_1 \oplus \mathbf{X}_2 \oplus \cdots \oplus \mathbf{X}_n \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times N_a \times d}$ and $\oplus$ is the stacking operation.

A convolutional layer contains N_c convolutional filters with different window sizes, and each filter has multiple convolutional kernels. We use the notation $\mathbf{w}_j \in \mathbb{R}^{h_j \times d}$ to denote j-th filters in this layer, where h_j means the filter is applied to a window of h_j attribute features to generate a new feature. The number of input and output channels of each filter is n and 24, respectively. The convolution stride and padding are fixed to 1 and 0, respectively. After the convolutional process, a max-pooling layer along the filter moving dimension is applied, yielding a 24-dimensional vector for each filter. The final outfit embedding, denoted as $g \in \mathbb{R}^F$, is obtained by concatenating these convolved vectors, where F is the dimensionality of the outfit embedding.

4.2.2 Multi-label Graph Convolutional Networks

We use Multi-label Graph Convolutional Networks [3] (ML-GCN) to train classifiers of the physical labels. ML-GCN is a graph convolutional networks (GCN) [8] based model which takes advantage of capturing the label correlations by treating the classifiers of labels as nodes. The adjacency matrix A is constructed based on the conditional probability of label L_j when label L_i appears as illustrated in Figure 4. The i, j entry of the matrix A is $A_{ij} = P(L_j|L_i)$, and matrix A is a weighted and asymmetrical matrix.

We briefly describe how ML-GCN is applied in this work. The generic layer-wise propagation rule of a GCN layer is:

$$x^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}x^{(l)}\Theta^{(l)}) \tag{2}$$

where $\widetilde{A} = A + I_D$ is the adjacency matrix of the graph with self-connections and $\widetilde{D} = \sum_j \widetilde{A}_{ij}$ is the degree matrix. $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix where N denotes the number of nodes in the graph. $x^{(l)} \in \mathbb{R}^{N \times C^{(l)}}$ is the matrix of activations in the l^{th} layer with $C^{(l)}$ feature maps. $\Theta^{(l)} \in \mathbb{R}^{C^{(l)} \times C^{(l+1)}}$ is the trainable weight matrix. $\sigma(\cdot)$ denotes the nonlinear activation function. $x^{(l+1)} \in \mathbb{R}^{N \times C^{(l+1)}}$ is the convolved feature matrix with $C^{(l+1)}$ feature maps.

A two-layer stacked GCN is selected to learn classifiers using the layer-wise propagation rule of Eq. 2. Taking the label representation with C physical labels $X \in \mathbb{R}^{N \times C}$ and the

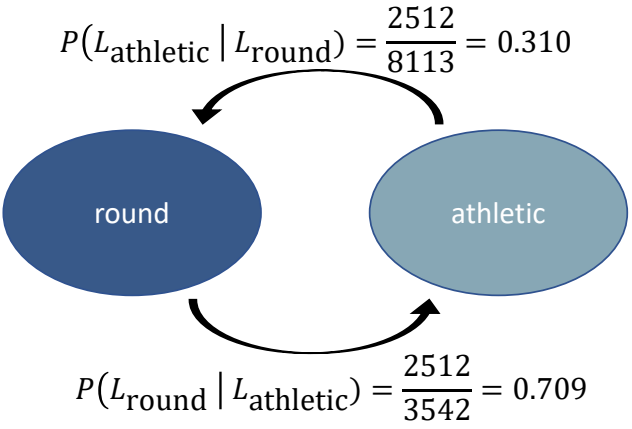

Figure 4: Construction of adjacency matrix based on conditional probability of two physical labels, *i.e.*, *round* and *athletic*. When *athletic* is not compatible with an outfit, there is a high probability that *round* is also not compatible with this outfit.

adjacency matrix $A \in \mathbb{R}^{N\times N}$ as input, a two-layer GCN model $f(X,A)$ can be expressed mathematically as:

$$Z = f(X,A) = \widehat{A}\,\text{ReLU}(\widehat{A}XW^{(0)})W^{(1)} \tag{3}$$

where $\widehat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$ is normalized version of adjacency matrix. $W^{(0)} \in \mathbb{R}^{C\times H}$ and $W^{(1)} \in \mathbb{R}^{H\times F}$ are two trainable weight matrices for the first and second layer, respectively and H is the dimension of the hidden layer. $Z \in \mathbb{R}^{N\times F}$ is the classifier matrix with F feature maps.

By applying label classifiers Z to the outfit embedding $\mathbf{g} \in \mathbb{R}^F$, the predicted score $\widehat{\mathbf{y}}$ is a non-parametric product of them:

$$\widehat{\mathbf{y}} = Z \cdot g \tag{4}$$

4.2.3 Objective Function

We evaluate the multi-label classification loss as follows:

$$L_1 = \sum_{n=1}^{N} \mathbf{y}^n \log(\sigma(\widehat{\mathbf{y}}^n)) + (1-\mathbf{y}^n)\log(1-\sigma(\widehat{\mathbf{y}}^n)) \tag{5}$$

where $\mathbf{y} \in \mathbb{R}^N$ is the ground truth physical labels of an outfit, and $\sigma(\cdot)$ is the sigmoid function. The overall cost function is defined as follows:

$$J(\Theta_{\text{FCN}}) = L_1 + \frac{\lambda}{2}\|\Theta_{\text{FCN}}\|_2^2 \tag{6}$$

where Θ_{FCN} is the trainable parameters of FCN and λ is the L2 regularization hyperparameter.

5 Experiment

Implementation Details. For the outfit encoder, the convolutional layer has five filters with different window sizes, *i.e.*, 1, 2, 4, 6, and 8. The number of fashion attributes N_a is 14. For

Table 2: Quantitative results on Body Shape attributes.

	Body shape							
Methods	top hourglass	hourglass	athletics	inverted triangle	triangle	spoon	round	dimension
Linear	**15.47**	63.90	66.47	76.14	63.41	63.09	71.96	70.90
ResNet [8]	10.73	31.76	33.37	79.22	67.86	67.15	66.44	65.26
Attn [23]	9.48	30.20	31.69	69.68	59.07	57.46	61.36	61.52
FCN	15.39	**66.53**	**70.15**	**83.48**	**70.29**	**69.82**	**77.52**	**76.35**

Table 3: Quantitative results on the rest attributes, excepting Body Shape.

	Skin			Hair color		Height		Breasts	Contrast
Methods	yellow	dark	brown	light brown	grey	high	low	big	low
Linear	11.59	24.35	14.35	9.17	13.31	17.38	13.94	31.23	12.27
ResNet [8]	12.05	41.57	14.29	**9.83**	**13.68**	18.08	11.98	26.50	12.06
Attn [23]	12.31	11.68	14.27	8.42	12.24	14.30	12.43	27.51	12.29
FCN	**13.24**	**46.84**	**15.11**	9.31	13.00	**21.57**	**23.23**	**31.91**	**12.72**

the GCN module, we use a two-layer stacked GCN and the output dimension of these two layers are 200 and 120. A pretrained *VGG* [20] is utilized as the attribute feature extractor. Both the height and width of the images are cropped to 224, and the dimension of the attribute feature vectors is 512. The main color feature extracted using FOCO system [29] is also added to the input feature maps. The physical labels are encoded by Glove [17] into 100-dimensional word embeddings. The FCN is trained in an end-to-end manner on the O4U dataset with a batch size of 10 on NVIDIA RTX 3070 GPU. We use SGD [18] as the optimizer with the learning rate, momentum, and weight decay are $1e^{-1}$, 0.9, and 5e-5, respectively. An exponential decreasing schedule for the learning rate and an early stop training strategy are both adopted.

Compared Approaches. 1. SVM [16]: The support vector machine (SVM) is chosen as one of the baselines to demonstrate the effectiveness of our approach. **2. Linear:** A network consisting of multiple fully connected layers and ReLU activation functions. **3. ResNet [8]:** We trained the ResNet with the input of the mean value of all item images. **4. Attention [23]:** We applied several stacked multi-head attention layers to encode an outfit with various attribute vectors into one vector.

5.1 Quantitative Results

Following the general practice [3, 22, 24], we report the performance of models on these metrics: mean average precision (mAP); average per-class precision (CP), recall (CR), and F1 (CF1); average overall precision (OP), recall (OR), and F1 (OF1). Average per-class metrics evaluate each label individually and then average over all labels. Average overall metrics evaluate over all examples. We also report the results of these metrics on top-3 labels.

We report mAP results for 17 physical labels in Tables 2 and 3. Our proposed method FCN achieves the best performance over 14 out of 17 labels compared with other baseline methods. Especially on labels belonging to the body shape category, our method achieves a huge improvement compared to other methods.

Model performances covering all 17 labels are reported in Table 4. FCN outperforms other baselines on most of all metrics. SVM, an effective machine learning method, shows good performance in terms of average overall metrics. However, FCN surpasses SVM by

Table 4: Main metrics on 17 physical attributes.

		All						Top-3					
Methods	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
SVM [16]	-	28.07	33.10	30.38	68.70	61.54	64.90	-	-	-	-	-	-
Linear	37.59	26.59	**33.93**	29.81	63.23	**65.14**	64.17	28.96	20.57	24.06	68.25	41.29	51.46
ResNet [6]	34.22	22.83	27.55	24.97	64.29	57.18	60.53	23.98	18.80	21.08	67.52	40.06	50.29
Attn [23]	29.76	18.18	29.41	22.47	61.82	62.33	62.07	11.44	17.65	13.88	64.82	39.22	48.87
FCN	**42.14**	**32.29**	33.84	**33.04**	**68.89**	62.17	**65.36**	**34.16**	**21.06**	**26.06**	**73.25**	**41.32**	**52.83**

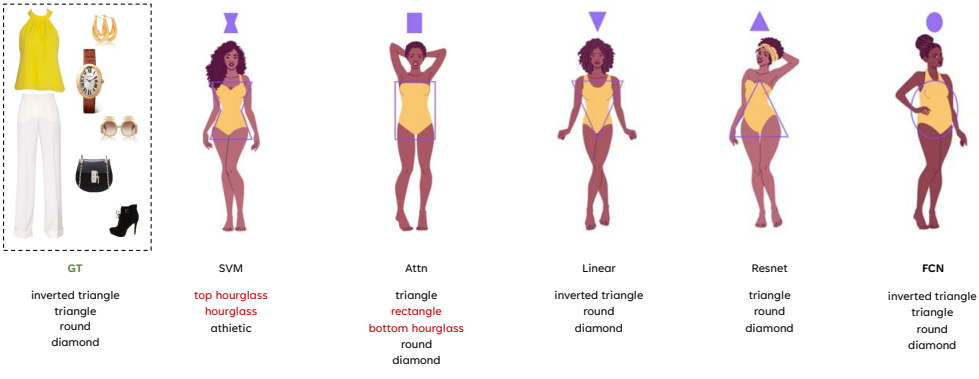

Figure 5: Qualitative results of all compared methods and the proposed FCN. The text in red is the wrong prediction. FCN precisely predicts all incompatible body shapes for the given outfit.

4.22, 0.74, and 2.66 on the CP, CR, and OP. The linear method works best in the recall indexes, indicating that this method may have a high sensitivity to the labels. The performance of ResNet is not good on mAP, and it indicates that treating outfits as the mean value of item images is not a good idea for this task.

5.2 Qualitative Results

Additionally, we present the qualitative results in Figure 5. People with body figures including "inverted triangle", "triangle", "round", and "diamond" is not suitable for the outfit on the left side. The main reason is that the silhouette of the tank top and the straight-line pants are not matched these types of body shapes. It can be seen that the FCN precisely points out all incompatible body shapes among the comparative methods and thus indicates that our method well learns the compatibility between fashion outfits and personal physical information.

5.3 Ablation Study

Effect of filter region size. We explore the sensitivity of different combinations of filter region size. As shown in Table 5, only using one kind of convolutional filter size shows the worst performance. Using filters with a big region size (relative to attribute number 14) has a negative effect on model performance. Using multiple filters with the same size achieves the best result on mAP and OF1, but the results are lower than FCN on the top-3 labels. The combination we use in FCN (1, 2, 4, 6, 8) shows the best performance on CF1 and Top-3 metrics.

Table 5: Effect of filter region size.

		All						Top-3					
Region size	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
(1)	40.68	32.55	32.99	32.77	67.68	62.50	64.99	29.56	20.27	24.05	71.29	40.53	51.67
(2)	38.93	28.36	32.42	30.25	68.67	61.28	64.77	30.64	20.44	24.52	73.01	40.24	51.89
(4,4,4,4,4)	**43.11**	**32.82**	33.46	**33.13**	68.70	62.02	65.19	32.30	20.82	25.32	72.30	40.87	52.22
(8,9,10)	41.38	28.29	32.72	30.35	68.83	61.29	64.85	30.29	21.19	24.93	73.12	40.96	52.51
(1,2,4,6,8)	42.14	32.29	**33.84**	33.04	**68.89**	**62.17**	**65.36**	**34.16**	**21.06**	**26.06**	**73.25**	**41.32**	**52.83**

Table 6: Effect of numbers of kernels for each filter.

		All						Top-3					
No. Kernels	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
2	35.46	26.54	35.58	30.40	62.52	68.25	65.26	27.94	19.72	23.12	66.03	39.95	49.78
12	41.67	32.19	33.91	33.03	67.96	63.09	65.44	**36.09**	20.77	**26.37**	72.31	40.95	52.29
24	42.14	32.29	**33.84**	**33.04**	68.89	**62.17**	**65.36**	34.16	**21.06**	26.06	73.25	**41.32**	**52.83**
48	**42.65**	**32.55**	33.10	32.82	**69.17**	60.90	64.77	34.75	21.02	26.20	73.75	40.78	52.52

Table 7: Effect of numbers of GCN layers.

		All						Top-3					
No. GCN	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
1	40.85	**34.11**	32.09	**33.07**	68.19	61.38	64.61	25.01	18.93	21.55	71.63	39.67	51.06
2	**42.14**	32.29	**33.84**	33.04	68.89	**62.17**	**65.36**	**34.16**	**21.06**	**26.06**	**73.25**	**41.32**	**52.83**
4	40.73	28.59	32.24	30.31	**69.05**	60.46	64.47	30.49	20.83	24.75	73.02	40.40	52.02
8	39.45	28.00	32.29	29.99	67.73	60.19	63.74	21.25	19.55	20.36	71.18	35.54	47.41

Effect of numbers of kernels for each filter We then explore the effect of the different numbers of kernels while keeping the filter region size to be the same and report the results in Table 6. We find that the performance achieves the best results when the number of kernels is 24. Using too few convolutional kernels will deteriorate performance significantly. Using too many kernels cannot dramatically improve performance while it causes a negative impact on recall metrics.

Effect of numbers of GCN layers We report the effects of different numbers of GCN layers in Table 7. We find that deeper multi-layer GCNs degrade the performance on almost all metrics. Therefore we choose to use a two-layer stacked GCN in our model.

6 Conclusion

We introduce a new task, Fashion Cognitive Learning, which targets to learn the compatibility among fashion outfits and personal physical information. The new framework named Fashion Convolutional Network is proposed to learn the relationships among visual-semantic embedding of outfit composition and appearance features of individuals. For implementation, we build a new large-scale fashion outfit dataset, O4U, covering comprehensive personal physical information. Extensive results demonstrate the advance of the proposed framework compared with all alternative methods. Expanding the dataset to different consumer groups accordingly is one of our future works.

Acknowledgements

This research is supported by the Laboratory for Artificial Intelligence in Design (Project Code: RP3-1) under the InnoHK Research Clusters, Hong Kong SAR Government.

References

[1] Imran Amed, Johanna Andersson, Achim Berg, Martine Drageset, Saskia Hedrich, and Sara Kappelmark. The state of fashion 2018: Renewed optimism for the fashion industry, 2017.

[2] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2662–2670, 2019.

[3] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019.

[4] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1078–1086. ACM, 2017.

[5] 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*, pages 770–778, 2016.

[6] Shintami Chusnul Hidayati, Cheng-Chun Hsu, Yu-Ting Chang, Kai-Lung Hua, Jianlong Fu, and Wen-Huang Cheng. What dress fits me best? fashion recommendation on the clothing style for personal body shape. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 438–446, 2018.

[7] Seongjae Kim, Jinseok Seol, Holim Lim, and Sang-goo Lee. False negative distillation and contrastive learning for personalized outfit recommendation. *arXiv preprint arXiv:2110.06483*, 2021.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. Hierarchical fashion graph network for personalized outfit recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–168, 2020.

[10] Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3311–3319, 2020.

[11] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Explainable fashion recommendation with joint outfit matching and comment generation. *arXiv preprint arXiv:1806.08977*, 2018.

[12] Yining Liu and Yanming Shen. Personal tastes vs. fashion trends: predicting ratings based on visual appearances and reviews. *IEEE Access*, 6:16655–16664, 2018.

[13] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. Learning binary code for personalized fashion recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10562–10570, 2019.

[14] Takuma Nakamura and Ryosuke Goto. Outfit generation and style extraction via bidirectional lstm and autoencoder. *arXiv preprint arXiv:1807.03133*, 2018.

[15] Charles Packer, Julian McAuley, and Arnau Ramisa. Visually-aware personalized recommendation using interpretable image representations. *arXiv preprint arXiv:1806.09820*, 2018.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[18] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 2007.

[19] Dikshant Sagar, Jatin Garg, Prarthana Kansal, Sejal Bhalla, Rajiv Ratn Shah, and Yi Yu. Pai-bpr: Personalized outfit recommendation scheme with attribute-wise interpretability. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 221–230. IEEE, 2020.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Omprakash Sonie, Muthusamy Chelliah, and Shamik Sural. Personalised fashion recommendation using deep learning. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 368–368.

[22] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[24] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[25] Xin Wang, Bo Wu, Yun Ye, and Yueqi Zhong. Outfit compatibility prediction and diagnosis with multi-layered comparison network. In *Proceedings of the 2019 ACM on Multimedia Conference*. ACM, 2019.

[26] Yufan Wen, Xiaoqiang Liu, and Bo Xu. Personalized clothing recommendation based on knowledge graph. In *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 1–5. IEEE, 2018.

[27] Heming Zhang, Xuewen Yang, Jianchao Tan, Chi-Hao Wu, Jue Wang, and C-C Jay Kuo. Learning color compatibility in fashion outfits. *arXiv preprint arXiv:2007.02388*, 2020.

[28] Haitian Zheng, Kefei Wu, Jong-Hwi Park, Wei Zhu, and Jiebo Luo. Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5014–5023. IEEE, 2021.

[29] Xingxing Zou, Wai Keung Wong, Can Gao, and Jie Zhou. Foco system: a tool to bridge the domain gap between fashion and artificial intelligence. *International Journal of Clothing Science and Technology*, 2019.

[30] Xingxing Zou, Zhizhong Li, Ke Bai, Dahua Lin, and Waikeung Wong. Regularizing reasons for outfit evaluation with gradient penalty. *arXiv preprint arXiv:2002.00460*, 2020.